Development of Features for Recognition of Handwritten *Odia* Characters

Tusar Kanti Mishra

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Development of Features for Recognition of Handwritten *Odia* Characters

Thesis submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science and Engineering

by

Tusar Kanti Mishra
(Roll: 511CS107)

under the guidance of

Prof. Banshidhar Majhi

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

September 2015

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

September 22, 2015

Certificate

This is to certify that the work in the thesis entitled **Development of Features for Recognition of Handwritten *Odia* Characters** by **Tusar Kanti Mishra** bearing roll number 511CS107, is a record of an original research work carried out under my supervision and guidance in partial fulfilment of the requirements for the award of the degree of **Doctor of Philosophy** in **Computer Science and Engineering**. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. B. Majhi

Dedicated to . . .

Maa, Bapa, Tiku, Lily and Silku

Acknowledgment

अज्ञानतिमिरान्धस्य ज्ञानाञ्जनशालाकया ।
चक्षुरुन्मीलितं येन तस्मै श्रीगुरवे नमः ॥

"Meaning: Salutations to the Guru who removes the darkness of ignorance from my blind (Inner) eyes by applying the collyrium of the light of knowledge."

I owe deep gratitude to the ones who have contributed greatly in completion of this thesis.

Foremost, I would like to express profound gratitude to my honorable supervisor, Prof. Banshidhar Majhi for his invaluable support, encouragement, supervision and useful suggestions throughout this research work. His moral support and continuous guidance enabled me to complete my work successfully.

I am thankful to Prof. Santanu Kumar Rath for his constant encouragement and support. His regular suggestions made my work easy and proficient.

I am grateful to Prof. Lambert Schomaker, who provided me with continuous support to carry out research in his laboratory for six months in the University of Groningen, The Netherlands.

I am very much indebted to Prof. Pankaj K Sa, and Prof. Ratnakar Dash for providing insightful comments at different stages of the thesis that were indeed thought provoking. My special thanks goes to Prof. Sarat Kumar Patra, Prof. Dipti Patra and Prof. Ashok Kumar Turuk for contributing towards enhancing the quality of the work in shaping this thesis.

I would like to thank all my friends and lab-mates for their encouragement and understanding. Their help can never be penned with words.

I am indebted to my father-in-law, late Paramananda Pati, who, in my absence took utmost care of my family till the last count of his breath. May his sacred soul rest in peace.

Most importantly, none of this would have been possible without the love, sacrifice and patience of my family. I would like to express my heart-felt gratitude to my family to whom this dissertation is dedicated to.

Tusar Kanti Mishra

Abstract

In this thesis, we propose four different schemes for recognition of handwritten atomic *Odia* characters which includes forty seven alphabets and ten numerals. *Odia* is the mother tongue of the state of Odisha in the republic of India. Optical character recognition (OCR) for many languages is quite matured and OCR systems are already available in industry standard but, for the *Odia* language OCR is still a challenging task. Further, the features described for other languages can't be directly utilized for *Odia* character recognition for both printed and handwritten text. Thus, the prime thrust has been made to propose features and utilize a classifier to derive a significant recognition accuracy.

Due to the non-availability of a handwritten *Odia* database for validation of the proposed schemes, we have collected samples from individuals to generate a database of large size through a digital note maker. The database consists of a total samples of $17,100$ ($150 \times 2 \times 57$) collected from 150 individuals at two different times for 57 characters. This database has been named Odia handwritten character set version 1.0 (OHCS v1.0) and is made available in *http://nitrkl.ac.in/Academic/Academic_ Centers/Centre_ For_ Computer_ Vision.aspx* for the use of researchers.

The first scheme divides the contour of each character into thirty segments. Taking the centroid of the character as base point, three primary features length, angle, and chord-to-arc-ratio are extracted from each segment. Thus, there are 30 feature values for each primary attribute and a total of 90 feature points. A back propagation neural network has been employed for the recognition and performance comparisons are made with competent schemes.

The second contribution falls in the line of feature reduction of the primary features derived in the earlier contribution. A fuzzy inference system has been employed to generate an aggregated feature vector of size 30 from 90 feature points which represent the most significant features for each character. For recognition, a six-state hidden Markov model (HMM) is employed for each character and as a consequence we have fifty-seven ergodic HMMs with six-states each. An accuracy of 84.5% has been achieved on our dataset.

The third contribution involves selection of evidence which are the most informative local shape contour features. A dedicated distance metric namely, far_count is used in computation of the information gain values for possible segments of different lengths that are extracted from whole shape contour of a character. The segment, with highest information gain value is treated as the evidence and mapped to the corresponding class. An evidence dictionary is developed out of these evidence from all classes of characters and is used for testing purpose. An overall testing accuracy rate of 88% is obtained.

The final contribution deals with the development of a hybrid feature derived from discrete wavelet transform (DWT) and discrete cosine transform (DCT). Experimentally it has been observed that a 3-level DWT decomposition with 72 DCT coefficients from each high-frequency components as features gives a testing accuracy of 86% in a neural classifier.

The suggested features are studied in isolation and extensive simulations has been carried out along with other existing schemes using the same data set. Further, to study generalization behavior of proposed schemes, they are applied on *English* and *Bangla* handwritten datasets. The performance parameters like recognition rate and misclassification rate are computed and compared. Further, as we progress from one contribution to the other, the proposed scheme is compared with the earlier proposed schemes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the ancient dreams ever is to enable a machine to replicate human functions, like reading and understanding. This dream is growing to reality since the last six decades with the evolution of readability efficiency of machines. During this evolution, optical character recognition (OCR) is one among the most successful technology in the field of pattern recognition and artificial intelligence. Now, there exist various commercial applications of OCR, still they are far behind the human reading capabilities.

Optical character recognition (OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text. It is widely used as a form of data entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method of digitizing printed texts so that it can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as machine translation, text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision. There exist numerous important applications of OCR systems. Some of these applications include robotics vision, cell phone text tools, preservation of ancient manuscripts, archiving official records, making aids for the visually challenged etc. OCR system is also used as prerequisites in some speech processing tasks where a textual image is converted into speech. Irrespective of the hardware arrangements, the recognition scheme plays a vital role in any OCR system. There are readily available OCR schemes for recognition of printed characters for numerous types of fonts and languages. As illustrated in Fig. 1.1, a general OCR begins with capturing of rough textual images using different input devices and ends in the post processing where the refined results are manipulated as per user requirements. Pre-processing, feature extraction, and classification are the most important phases. Pre-processing schemes used for a character image are mostly standardized and are quite capable of resolving challenging tasks. Major challenges lie in development of features and feature reduction schemes for effective recognition. These challenges get doubled while dealing with handwritten character recognition.

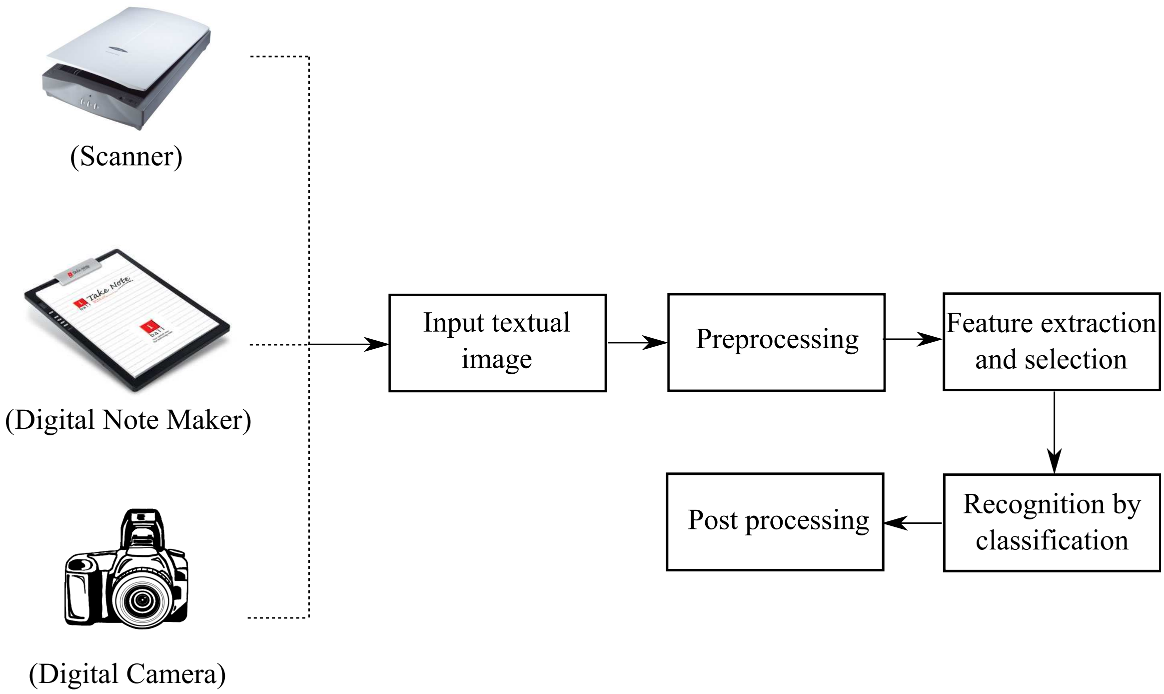


Fig. 1.1 – General overview of an OCR system.

The importance of OCR is increasing day by day in accordance with the increase in the number of its applications. Few important applications are,

1. scanning and recognition of official documents,
2. license plate recognition,
3. extraction and manipulation of textual data from maps,
4. development of reading schemes for the illiterates and visually challenged persons,
5. robotics vision,
6. development of cell phone application related to handwriting recognition, and
7. safe preservation of ancient documents by digitization techniques.

Now a days, OCR applications are developed with two main approaches. A group of applications is aiming for the on-line character recognition, whereas, others target the off-line recognition. On-line recognition schemes deal with simultaneous capturing of text and producing corresponding recognized character. On the other hand, off-line recognition schemes adhere to time independence whereby they do the recognition job separately in a sequence or they aim at recognizing stored character sets.

1.1 Steps in an OCR

As shown in Fig. 1.1, OCR system begins with capturing of rough textual images using different input devices and ends in the post processing step where the refined results are manipulated as per user requirement. The steps involved in an OCR are discussed below in brief.

- **Input Textual Image**

  In OCR, the system accepts the characters, words, or sentences as an image by input devices like scanner, camera, and note-maker. There exist several issues like noise, skewness, and variability in size associated with these acquired images. Hence, these input images need to be pre-processed prior to the feature extraction and recognition of characters.

- **Preprocessing**

  Preprocessing deals with the refinement and standardization of input textual images. Numerous image processing operations like noise removal, skew correction, word/character segmentation, atomic character extraction, slant correction, thinning, and dilation are performed on the input images.

  Initially, an input image is converted to a standard size followed by binarization. This helps in bringing the image into two tones from its colored version. To avoid distortion and breaks within individual characters, a thickening operation is performed to generate a uniform image without breaks. Subsequently, if required, a thinning operation is applied to bring the edges to a single pixel width.

- **Feature extraction and selection**

  Every entity has its own properties which uniquely identify it. The identification and extraction of suitable features is the most important step in OCR. It is carried out with minimal error as well. The more is the discriminating nature a feature is, the more is its suitability. The general taxonomy for feature extraction techniques in the context of OCR is given below.

(a) **Correlation:** The features, based on correlation, takes into consideration the distance metrics. They seek a point-wise analysis of the input character image. These kind of features are efficient only when there are infinitesimal noise present in the input samples. These features are simple to compute and provide a good understanding of the data.

(b) **Transformation:** This process involves alteration and transmission of input character sample in term of its representation. The axes and description parameters are altered to another domain of representation. The features based on transformation are robust, tolerant to noise and distortion. The transformation activities are capable of minimizing the dimensionality of the feature sets. The wavelet and Fourier transforms are the popular transformations which have been used for OCR.

(c) **Statistical:** This method extracts features from the statistical description of characters. The advantage of such features is that they execute faster with less computational cost. They are also invariant to change in font type. Popular examples include zoning techniques and moment based feature extraction.

(d) **Geometric:** Shapes of the characters are the main target for this technique. The strokes, curvilinearity, line segments, directional attributes, and their relativities are taken into account. They are mostly invariant to distortion and color changes.

- **Classification**

  Classification involves identification of an observation. It determines the belongingness of a probe character to a particular class. This is carried out on the basis of existing training data that contains some samples for which the class-label is already known. There exists the neighborhood technique that recognizes the current sample of observation on the basis of its neighborhood known samples. Several state-of-the-art algorithms have been proposed [1–4] for the purpose. The classification based on neighborhood approach does not demand any prior knowledge. They are executed with less computational cost. Some classification algorithms use statistical information on input patterns. The Bayesian classifier and the Parzen window classifiers are good examples in this context. The recent and popular classifiers are support vector machine (SVM) [5], and artificial neural network (ANN) [6]. These type of classifiers emulate the function of the human brain for pattern recognition. In the training phase, a set of input-output relationship is learned. The recognition by classification phase in an OCR tries to achieve good recognition rate and faster execution.

1.2 Related Works on OCR

Turing is assumed to be the first person who made an effort to make an aid for the visually challenged [7]. OCR for printed characters was performed using template matching, on the other hand, binarized image samples, and statistical classifiers are used for hand-printed character recognition [8–10]. A survey on character recognition techniques used prior to 1980 is reported in [11]. During the 80's, emphasis are put on structural feature in combination with the statistical features [12–15]. On-line OCR using structural features is reported in [16] where the sequence of points following the trend of instantaneous writing are extracted which includes curves and line segments.

Image processing and pattern recognition techniques have been considered to be in the domain of artificial intelligence. Efficient classifiers like fuzzy reasoning, artificial neural networks (ANN), and hidden Markov model (HMM) are developed and used in OCR systems. HMM evolved as a key technique for speech recognition. In the past few years, several HMM-based OCR techniques have been reported in the literature. In [17, 18], OCR systems for Arabic characters have been proposed which utilize sub-character HMM modeling efficiently. The variation in the shapes of characters with respect to positions are extracted in order to model a compact system with reduced set. The derivative features extracted by running a sliding window protocol on character images has also been used. They have referred the IFN/ENIT benchmark dataset of handwritten Arabic texts for validating the scheme. Though the recognition rate is not so high (85.12%), the method is a good example of handwritten OCR. In [19–21], over-splitting of an image into overlapping segments is performed. They use dynamic programming approaches for the recognition and segmentation process.

In the last two decades, many of the OCR systems have adopted the concept of recognition based on segmentation [22]. Techniques, such as segmentation of lines in a document, segmentation of words from the lines, and segmentation of atomic characters from those words are followed in sequence. Subsequently, feature extraction and classification techniques are used for the recognition task. But the case of recognizing low-resolution and distorted characters is different. Although sophisticated techniques have been proposed [23, 24], still it has been a challenging task till date.

In [25], alpha-numeric *Bengali* character recognition is made using curvature feature. This work is supposed to be an initiative that use curvature feature for OCR in Indian script. Along with a *shiro-rekha* (horizontal line segment at the top of each character), the *Bengali* characters consist of strokes and vertices (junction points). These two important characteristics have been exploited in this work. They have concentrated on three features: count of points in the curvature maxima and minima, the points of inflexions where there occur a change from positive to negative curvature and vice-versa. They have used two neural networks that execute in sequence for the classification. In [26], the *shiro-rekha* found in *Bangla* texts are used as the reference for skew correction and segmentation. In this work, printed *Bangla* characters are divided into three distinct zones with reference to the positioning of head-line and base-line of the character. The key features used in this work are: bounding-box-width, rate of border-pixel, and curvature per unit width of the character. These features have been used in a sequence separately to arrive at a final recognition label. A benchmark dataset of handwritten *Bangla* compound characters has been provided by Das et al. [27]. The dataset contains $55,278$ samples. Another database for *Bangla* OCR has been reported in [28], that contains $37,858$ samples. In [29], skeletal-convexity is used for recognition of both printed and handwritten *Bangla* characters. A survey on *Bangla* and *Devanagari* character recognition is reported in [30].

The *Devanagari* character set is used for writing *Hindi, Sanskrit, Marathi, and Nepali* languages. More than 700 millions of people across the world with a majority of users from India and Nepal are using this language. In [31, 32], OCR schemes for *Devanagari* scripts have been proposed. These works intend to recognize handwritten numerals and constrained handwritten alphabets. The four-directional (top, bottom, left, and right) line segments are extracted along with intersection points and are used as the basic features. Decision tree classifier is used for recognition in these works. Other early works on this script are reported in [33, 34]. In [35], the researcher has proposed a method for atomic *Devanagari* character recognition. He has used the quad-tree data structure. Atomic character is divided into four quadrants at a regular interval along axes. The numbers of active pixels (ON state) are recorded into the feature matrix for each division. Structural features are used in [36], where, existence and the reference position of top horizontal bars and character strokes are used as features for the purpose.

Compound handwritten *Devanagari* character recognition technique has been proposed in [37]. The authors have reported a maximum of 15% presence of joint characters in the scripts. Zernike moments, being rotation invariant, act as the key features for this work. They have made a comparative analysis between SVM and KNN (K-nearest neighbor) classifiers with the same feature sets extracted from $27,000$ samples. The SVM is found to be better than KNN in their experiment. A survey on *Bangla* and *Devanagari* OCR is reported by Bag et al. [30].

The *Kannada* language is used by over 50 million people from the south-Indian state of Karnataka and neighbors. Wavelet features from character contours are used in [38] for online *Kannada* OCR with ANN classifier. The same has also been applied on *Telugu* character dataset. Zernike moments and Hu's moments have been utilized in [39] for atomic machine-printed *Kannada* OCR along with ANN classifier and a recognition rate of 96.8% has been reported. Recognition technique for unconstrained *Kannada* handwritings has been discussed in [40]. The technique uses 2D-FLD (Fisher discriminant analysis) features. Comparative analysis has been attempted using different distance metrics during classification. In [41], an on-line OCR has been discussed using curvature features, where KNN and SVM are used for classification. Zone based recognition technique has been proposed in [42]. Each character image is divided into 64 zones (8×8 pixel size). Crack code i.e. the edge between foreground character pixels and background pixels are extracted for every zone. A total of $24,500$ samples (500 samples per character) are considered for the experiment with a character being represented by feature vector of size 1×256. Finally, multi-class SVM is used for classification. Recognition rate of 87.24% is achieved with k-fold ($k = 5$) cross-validation check.

Tamil is one among the oldest languages of India and is used in the state of Tamilnadu. Early work on machine-printed *Tamil* OCR is reported in [43]. Character images are represented using binary matrices. These matrices are encoded to strings to form the feature vectors. Simple string matching with stored dictionary strings is performed to recognize a test character. Another early work on handwritten *Tamil* OCR is reported in [44]. The researchers have used labeled-graphs for representing the structural compositions of input images. They perform the recognition by simply correlating these graphs with previously stored graphs of some fixed symbols. The topological matching procedure is used for computing correlation coefficients. A unique octal graph approach has been discussed in [45] for off-line *Tamil* OCR. Matching is performed by ranking the test samples with reference to their octal graph distance to previously stored templates. In [46], decision tree classifier is efficiently used for the *Tamil* OCR. The constituent components (lines, curves, loops) of a character are considered as basic features in this work. Local SIFT (scale invariant feature transformation) features have been utilized for off-line *Tamil* character recognition [47]. The LBG (Linde Buzo and Gray) algorithm has been used to construct a code book to reduce the retrieval time. Classification is carried out using k-means clustering. Recognition rate of 87% has been reported in this context.

Curvature feature based off-line *Odia* OCR has been proposed by Mohanty et al. [48]. The researchers have put emphasis on the curve like strokes present in the characters. Normalized input character images are divided into 49×49 blocks. The bi-quadratic interpolation technique is adopted to extract the curvature with three levels of quantizations. Further, direction of gradient followed by strength of gradient have also been taken into consideration. They have used the principal component analysis (PCA) for reducing the dimension of feature vectors. In [49], curvelet transform is used for multi-font *Odia* character recognition. The curvelet features are mentioned to be better in comparison to wavelet features. In [50], moment features and geometrical properties are considered for the purpose of recognition. They concluded that moment features are resistant to noise and font variations. In [51], Fisher ratio (F-ratio) based technique has been proposed for recognition of similarly shaped *Odia* characters. F-ratio is the ratio between the inter-class variance and intra-class variance. It has been used to assign weights to similar shaped characters. It reduces the similarity between two similar shaped characters and simultaneously it highlights the distinguishable parts between them. Unconstrained handwritten *Odia* numerals recognition using contour features has been proposed in [52]. Directional chain code histograms are computed among several blocks from a numeral image. These histograms are treated as features for the purpose of recognition. ANN has been used as the classifier in this work. The size of the dataset used in this work is $3,850$. Isolated handwritten *Odia* numeral recognition based on a concept of water reservoir has been proposed [53]. Topological and structural features have also been taken into consideration in this work. The parameters like number of reservoirs, their size, heights and positions, water flow direction, number of loops, centre of gravity positions of loops, and the ratio of reservoir/loop are used as feature vectors. The dataset used in this work consists of $3,550$ samples.

1.3 Motivation

An overall analysis on several works on OCR in general reveals that, contributions made on Indian languages (specifically *Odia*) are quite small in number. In many a cases, it is found that the structural and geometric features (shapes, loops, line segments, strokes, curves, etc.) have been extensively used. The inter-relationship among these features have also been considered for the purpose. These features have been pretty well exploited for designing machine-printed character recognizers. There are a number of Indian languages where such features can be utilized. This is because, the structure of characters in these languages hold such similar properties like curvilinear segments, line segments, and junction points. Many works in some Indian languages have also been carried out using these features. The statistical features have also played a significant role in resolving ambiguities in pattern analysis.

In the context of Indian OCR, both global and local features have been used for printed character recognition and very few works have been proposed for handwritten character recognition. The need for proper feature extraction schemes is clear and unambiguous for higher recognition accuracy.

For the classification task, almost every classifiers have been used in developing OCR for Indian languages. The use of neural network based classifiers have out-performed their counter parts, never the less, the HMM too, especially for Indian languages. Sometimes, the non-availability of good features have compelled the classifiers to backtrack.

1.4 Objectives

The objectives laid down in this thesis are to

- develop a database of sufficient samples of handwritten *Odia* characters collected from variety of users and make it available publicly,
- exploit the geometric features of *Odia* characters for developing practically viable recognition schemes,

- introduce robust local features that should be invariant to scaling and rotation, and
- use the image transformation schemes like DCT and DWT on *Odia* characters to develop hybrid energy features.

1.5 Organization of the Thesis

The overall work in this thesis is organized into six different chapters with four contributions in addition to introduction and conclusion. The contributions are discussed below in nutshell.

Database creation and shape primitive feature extraction: In **Chapter 2**, creation of the first-of-its-kind handwritten *Odia* dataset containing atomic characters is presented. A shape contour based approach has been presented for the recognition of handwritten *Odia* characters. Three primitive features have been proposed in this context. A back-propagation neural network (BPNN) has been adopted for classification. In addition, the scheme has been validated on handwritten *English* character set and *Bangla* numerals. Comparative analysis of the proposed scheme with three recent schemes reveals that the proposed features yield a better recognition rate.

Aggregated feature generation using fuzzy inference system and classification using HMM: A feature aggregation techniques using FIS (fuzzy inference system) has been presented in **Chapter 3**. The three primitive features so obtained in **Chapter 2** are aggregated into a single feature vector of smaller dimension using this technique. An HMM approach is followed to develop individual models for the characters of each class. These models have been used for predicting the label of a test character. The overall scheme is dubbed as AFHMMC (aggregated feature vector with HMM recognition). Simulation results show an improvement when compared to the scheme suggested in Chapter 2 in terms of recognition rate over all the three datasets.

Evidence collection based local feature extraction scheme: An efficient local shape descriptor based recognition scheme with a novel distance metric (far_count) has been proposed in **Chapter 4**. This chapter puts emphasis on the local shape features of a character. The scheme efficiently generates an evidence dictionary which contains meaningful local segments of the shape contour of characters. These evidence are used for recognizing the class label of an isolated test character. The scale and rotation invariance properties of these segments make the scheme more robust. Proper test has been conducted to evaluate the efficiency of the scheme. Comparative analysis with other competent schemes also gives satisfactory performance favoring the proposed scheme.

Development of hybrid energy feature using DCT and DWT: A novel hybrid feature based on DWT and DCT has been presented in **Chapter 5**. To decide the optimal number of DWT decomposition and number of DCT coefficients in each sub-band image, exhaustive simulation has been performed. Based on the experimental results, a neural classifier (BPNN) is trained to be used during testing. Recognition performance of the scheme is compared with existing competent schemes. It is in general (for the three datasets) observed that the proposed hybrid feature outperforms others. The scheme provides high accuracy and recognition rates.

1.6 Summary

The contributions are elaborated in different chapters along with simulation and results. Whatever necessary, the literatures on related works are also described.

Chapter 2

Contour Features for *Odia* Handwritten Character Recognition

It is observed that most of the OCR schemes reported in the literature perform well for a particular language whereas they perform poorly for other languages. This is due to the presence of inter-variations among characters in terms of shape and orientations. Hence, there exists a potential need of devising a general scheme for handwritten character recognition in multiple languages.

In this chapter, an efficient recognition scheme based on the shape contour information of character images has been proposed for handwritten *Odia* characters. Using polygonal approximation, the shape contour of a character image is divided into fixed number of segments (arcs). Keeping the centroid of the character as the origin, it is segmented into equidistant arcs. Subsequently, primitive features namely, length of the extension from the origin to the arc (l), angle between this extension and its corresponding chord(θ), and arc-to-chord-ratio (r) are extracted for each arc. These features are fed to a back propagation neural network (BPNN) for the purpose of recognition by classification. Simulation has been carried out in a large handwritten *Odia* data set and comparative analysis with other features has been made with respect to recognition accuracy. The proposed scheme is evaluated for character recognition on datasets of two other languages, namely *Bangla* and *English*. The scheme gives satisfactory performance results on the three languages.

2.1 Handwritten *Odia* Database Creation

To validate any scheme, it is necessary to use a standard dataset of significant size. Since very little work have been reported for handwritten *Odia* characters, no such standard dataset is available for this language. Only for handwritten *Odia* numerals, a scanned numeric-characters dataset of 2,000 samples is made available by the Indian Statistical Institute (ISI Kolkata, India) [54]. To validate the proposed scheme, we have faced difficulties and as a part of our research, a database of handwritten *Odia* characters has been created.

The preliminary requisite for any character recognition system is the acquisition of input character images and subsequently pre-processing them. These two phases of any OCR are vital and should be carried out with utmost care. This is to avoid the influence to the result due to flaws or distortions in the input dataset. This may result in a drastic decrease in the rate of recognition or a rise in the computation time involved. Keeping this in mind, the utmost care has been taken during the sample collection process.

A pen tablet is used for the purpose due to its advantages over traditional scanner. The characters collected are devoid of noise due to dust, liquid spilling etc. The sample collection form used for the purpose of data collection is shown in Fig. 2.1. It contains 57 atomic characters. Handwritten samples are collected two times from each user. Filled data sheets are collected from the scholars of different labs of the institute, and from people outside the institute. A total of 150 different users have been approached to give handwritten samples resulting to 300 samples for each character with a total of $17,100$ samples. The dataset so created is named as *Odia* handwritten character set (OHCS 1.0).

2.1.1 Preprocessing

We can see in Fig. 2.1 that a fixed block of uniform geometric space has been allocated to each character. Based on this block specification, the individual samples are extracted from the digital page and are stored as individual image. Each character image in the database is labeled as OH_$\mathrm{C}_i\mathrm{S}_j$ for i_{th} class ($(1 \leq \mathrm{i} \leq 57)$) and j_{th} sample ($1 \leq \mathrm{j} \leq 300$). For example, an image labeled "$\mathrm{OHC}_{10}\mathrm{S}_{15}$" is to represent the 15^{th} sample of the 10^{th} character class. Each character image is standardized to a size of dimension 64×64 (pixels) and subjected to standard image pre-processing techniques like Otsu method for binarization, interpolation, branch points application, dilation, thinning, and skeletonization [55]. Each pre-processing step has its significant role. The Otsu thresholding method is applied to convert the gray scale image into the corresponding binary equivalent image. As the characters are handwritten, there are possibilities of mild broken lines in their stroke paths. Interpolation is applied to remove the breaks if any in the character. This is followed by the branch points operation which gives node-corrected components for a character image. Dilation and

Fig. 2.1 – Scan copy of a sample *Odia* handwritten dataset collection page.

thinning are applied in sequence to draw uniformity in the thickness of the characters. Skeletonization is performed to get images with m-connected components. A final touch of thickening of 5-pixels width is applied to each of the character images. Now, the input sample is said to be ready for the feature extraction phase. The outputs after various steps for *Odia* character ବ ('ba') is shown in Fig. 2.2. The pre-processed character image is subsequently used for feature extraction and recognition.

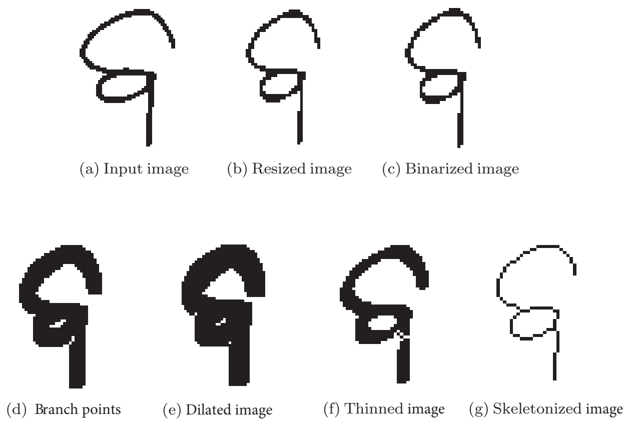


Fig. 2.2 – Example of several sequence outputs in the pre-processing phase.

2.2 Contour based Features with Neural Classification

The proposed scheme deals with extraction of contour-based features from each handwritten character followed by classification using a neural network classifier. Hence, the scheme is coined as contour-based features with neural classification (CFNC). Three different primary features namely, length(l), angle (θ), and chord-to-arc-ratio (r) are extracted from each character image. The overall scheme is depicted in Fig. 2.3. For a better understanding, the overall processes are elaborated below.

Each character image is represented as a 64×64 (pixels) image. In the pre-processing phase, several tasks like noise removal, standardization and normalization are performed on the image prior to feature extraction. Each character is represented by a one-dimensional shape contour descriptor $T = (t_1, t_2 ..., t_m)$ which is an ordered set of m real-valued variables taken clockwise from the contour. Polygonal approximation is applied to the contour descriptor to generate three basic feature-vectors for representing the character image.

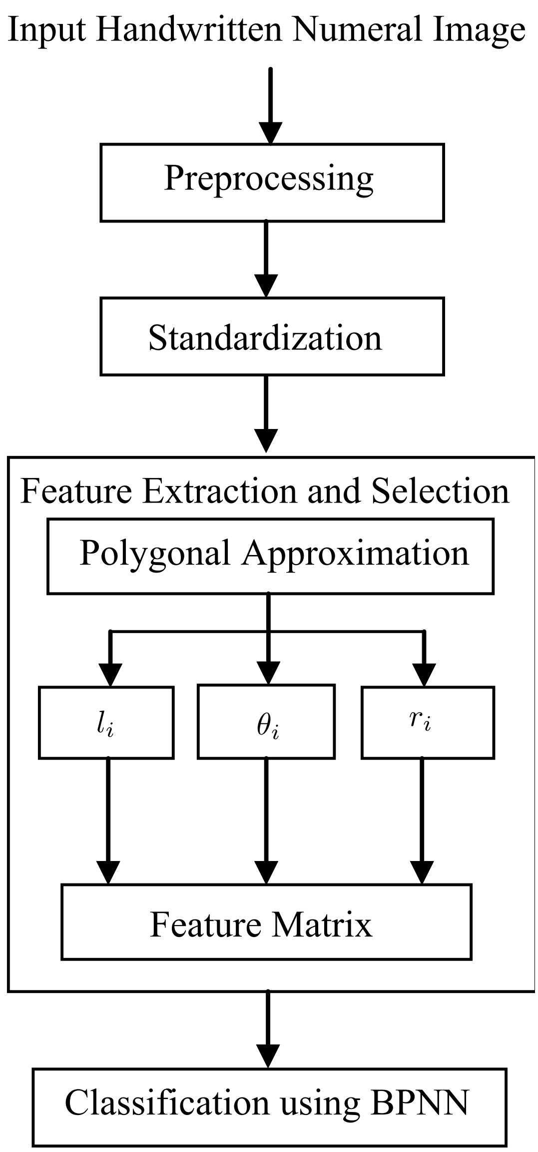


Fig. 2.3 – Overview of the proposed CFNC scheme.

Polygonal approximation has been applied on the pre-processed character image where, each character shape contour is segmented into S different segments, keeping the origin at the centroid of its shape. The contour segmentation process for *Odia* character ଅ ('ah') is shown in Fig. 2.4. For choosing the starting pixel in the profile in order to segment it into arcs, one can choose the farthest pixel or nearest pixel from the centroid. If more than one point exists satisfying the criteria (*e.g.* in case of circular character), then any of these two points can be selected. The top-left corner point on the shape contour is chosen as the starting point for segmentation. The steps followed to generate the features are listed in **Algorithm 1**. The algorithm for feature point selection has a time complexity of $O(S)$, where S is the number

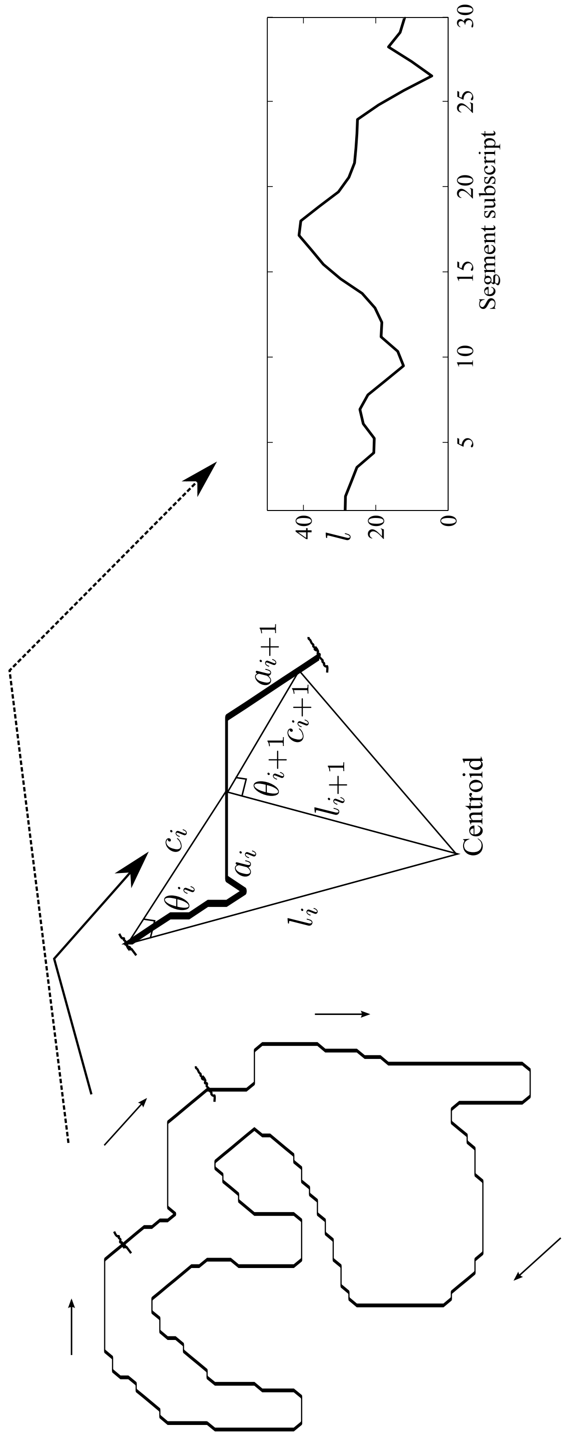


Fig. 2.4 – Illustrating contour segments of *Odia* character 'ah'.

of segments taken from the contour. The selection of number of segments is a heuristic choice, and it should be selected in such a manner where the average smoothness factor, r (ratio between chord and arc) should be more than 75%. This is to keep the smoothness of the perfect polygonal approximation to the contour description. For this case, S has been taken to be 30. Three primitive features namely, distance (l_i), angle (θ_i), and chord-to-arc-ratio (r_i) are extracted from these segments. The distance feature (l_i) represents the distance between the centroid and the starting point of i^{th} segment, the angle feature, θ_i is the angle between l_i and chord c_i lying between i^{th} and $(i+1)^{th}$ segment, and the ratio, r_i represents the ratio between the chord c_i and arc a_i. Thus, we have $1 \times S$ dimension of each primary feature and if we arrange the three different features in rows, we generate a feature vector f_v of dimension $3 * S$. These feature descriptors are unique to a particular character. A typical feature vector for the *Odia* character ଘ is given as, $f_v = \{l_1, l_2, \ldots, l_{30}, \theta_1, \theta_2, \ldots, \theta_{30}, r_1, r_2, \ldots, r_{30}\}$.

Algorithm 1: Shape_Contour_Feature_Extraction

Data: *Odia* handwritten character dataset

Result: feature vector for all characters

```
for each character image c_j in the dataset;          // 1 ≤ j ≤ |dataset|
do
    Initialize, f_j ← Null; l_j ← Null; θ_j ← Null; r_j ← Null;
    i ← 1;
    while i ≤ S ;                                     // S = number of segments
    do
        Compute l_i = length from centroid to the i^th segment of character c_j;
        Compute θ_i = angle between l_i and the chord corresponding to i^th
          segment;
        Compute r_i = ratio between the i^th segment and its corresponding
          chord;
        l_j = l_j||l_i; θ_j = θ_j||θ_i; r_j = r_j||r_i;
        i = i+1;
    end
    f_j = f_j||l_j||θ_j||r_j;
end
```

2.2.1 Recognition by classification using BPNN

Neural network has been successfully utilized for pattern classification and recognition. Generally, a neural network consists of a number of nodes and a set of associated links. These nodes are resembled as neurons, and the links describe the connections and data flow between these neurons. Connections are quantified by weights. These weights can be adjusted dynamically while training the network. A set of training instances is given during the training phase. Each training instance is particularly described by a feature vector/input vector. It should have an association with a desired output. This output is encoded as another vector and is called as the desired output vector.

Algorithm 2: Conjugate_Gradient_Algorithm (CGA)

1 Choose initial weight vector, w_1;
2 Set $p_1 = r_1 = -E(w_1), andk = 1$;
3 Calculate second order information, $E^H(w_k)p_k$ and $d_k = p'_k(s_k)$;
4 Calculate step size, $\mu_k = p'_k r_k and \alpha_k = \frac{\mu_k}{d_k}$;
5 Update weight vector, $w_(k+1) = w_k + \alpha_k p_k andr_(k+1) = E'(w_(k+1))$;
6 **if** $k \ mod \ m = 0$ **then**
7   restart algorithm: $p_(k+1) = r_(k+1)$ `// where` m `is the number of weights`;
8 **else**
9   create new conjugate direction, $\beta_k = \frac{(r_(k+1)^2 - r_(k+1)' r_k)}{\mu_k}$;
10     $p_(k+1) = r_{(k+1)} + \beta_k p_k$
11 **end**
12 **if** *the steepest descent direction* $r_k \neq 0$ **then**
13   set $k = k+1$ and go to step - 2;
14 **else**
15   return $w_{(k+1)}$ as desired minimum and terminate;
16 **end**

For the recognition purpose in our case, we used a feed-forward neural structure for classification of the characters. A set of feature vectors f_j corresponding to a character c_j with a target class t_j, the pair $(f_j : t_j)$ constitute the training pattern. Similarly, we collect the training patterns for each character in the dataset. For training, the back propagation based on conjugate gradient algorithm (CGA) is used and the weight update equations for this are defined as,

$$E_{qw}(y) = E(w) + E'(w)^T y + \frac{1}{2} y^T E''(w) y \tag{2.1}$$

$$E'_{qw}(y) = E''(w)y + E'(w) \tag{2.2}$$

where, $E(w)$ is the error function, E_{qw} is the quadratic approximation to E in a neighborhood of point w. The set of steps involved in CGA is listed in **Algorithm 2**. Using a step size scaling mechanism, it avoids time-consuming line search in every iteration of the training and achieves faster convergence.

2.3 Experimental Evaluation

To validate the proposed CFNC scheme, simulation has been carried out with handwritten characters of the *Odia* language. The overall simulation is divided into two different experiments to study various aspects of CFNC scheme. The experiments are discussed below in detail.

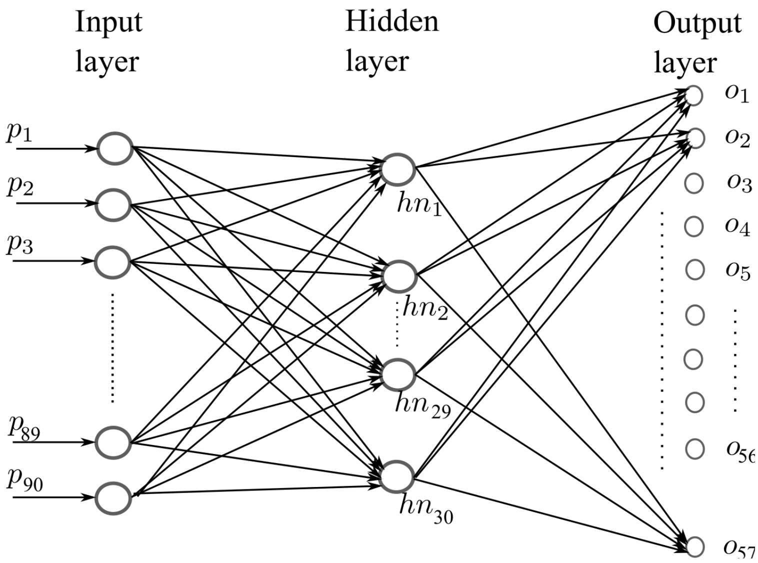


Fig. 2.5 – Neural network structure for training.

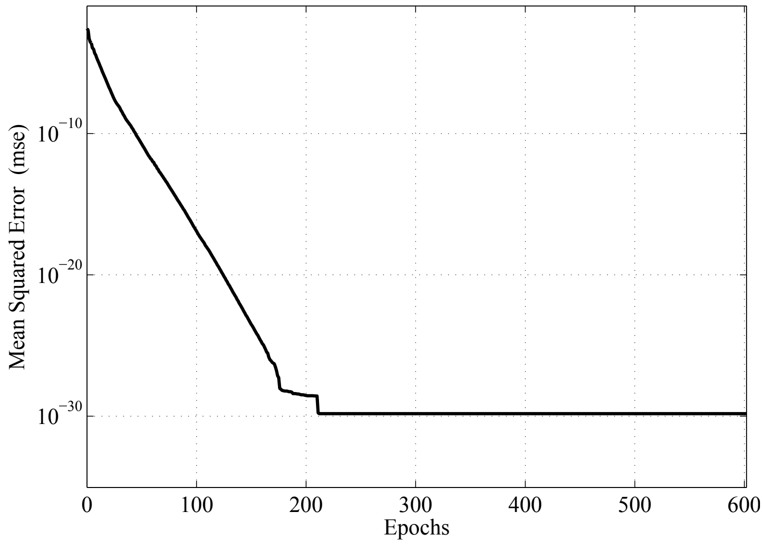
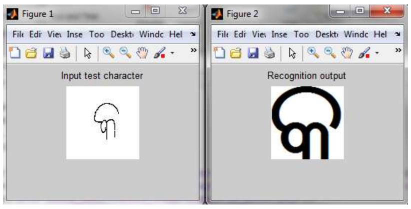


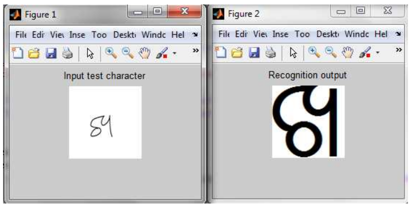
Fig. 2.6 – Convergence characteristic of the neural network.

2.3.1 Study of Training Convergence Characteristics

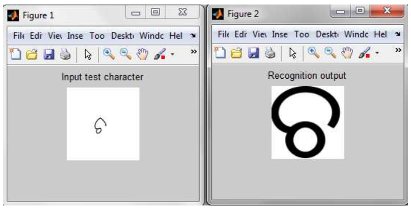
The feature vector f_j, for each sample character c_j is extracted using **Algorithm 1** and so the corresponding training pattern $f_j : t_j$. A total of 30 segments are chosen for each character, and hence the feature vector is of dimension 1×90 consisting of distance (l_j), angle (θ_j), and ratio (r_j)values in succession. For experimental evaluation, a set of 100 samples are selected randomly from each of the 57 classes and the total of $5,700$ training patterns, $(f_j : t_j)$ are used for training. The neural network structure used for the case of the OHCS dataset with the node specification as $90-30-57$ is shown in Fig. 2.5. The number of neurons employed in the hidden layer is experimentally determined to be 30 for faster convergence. The number of output neurons are decided according to the number of output labels. Hence, the neural structure is of $90-30-57$. To enhance the reliability and faster convergence of back propagation training, scaled conjugate gradient is used. The training convergence characteristics is shown in Fig. 2.6.



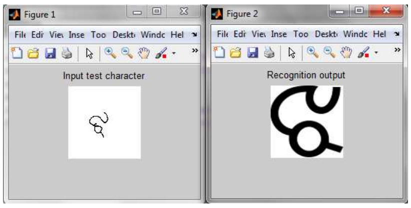
(a) Correct recognition of *Odia* character 'ka'



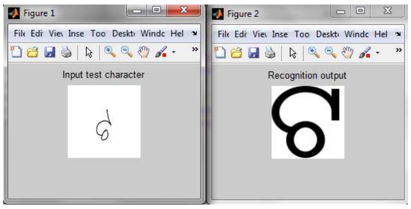
(b) Correct recognition of *Odia* character 'kha'



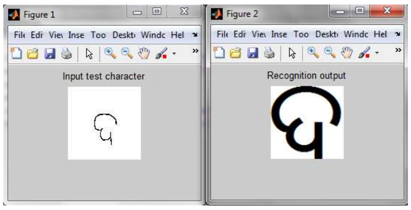
(c) Correct recognition of *Odia* character 'cha'



(d) Correct recognition of *Odia* character 'jha'



(e) Correct recognition of *Odia* character 'ta'



(f) Correct recognition of *Odia* character 'da'

Fig. 2.7 – Test run instances of the CFNC scheme.

2.3.2 Performance Analysis on *Odia* Characters

A total of two hundred handwritten characters for each character class have been selected randomly from the OHCS dataset which are not used during training and a total of 11,400 characters are used for testing. Overall accuracy is computed using the k-fold ($k = 10$) cross-validation [56]. Accuracy comparison of the proposed scheme has been made with state-of-the-art approaches namely, Zernike moments, curvature feature, and skeletal convexity. Overall accuracy comparison is shown in Table 2.1.

It is observed that, for all characters, the proposed CFNC scheme outperforms other competent schemes. Further, the simulation has been extended to other two languages *i.e.* *English*, and *Bangla* with available datasets [57, 58]. The proposed CFNC scheme outperforms other competent schemes in terms of rate of accuracy for these languages as well. Some output instances are shown in Fig. 2.10. The receiver operating characteristics curves (ROC) for *Odia* data is shown in Fig. 2.8.

Even though, most of the characters are recognized correctly (Fig. 2.10), it has been observed that there are few groups of characters which are similar in shape as shown in Fig. 2.9. Hence, for such groups, the CFNC scheme misclassifies one character as the other. It is evident from the confusion matrices of ପ ('pa') and ୨ (two) as given in Table 3.15, and Table 3.15. The character ୬ (six) is found to be misclassified as ୭ (seven) for the maximum time followed by ଅ ('ah') which is misclassified as ଥ ('tha').

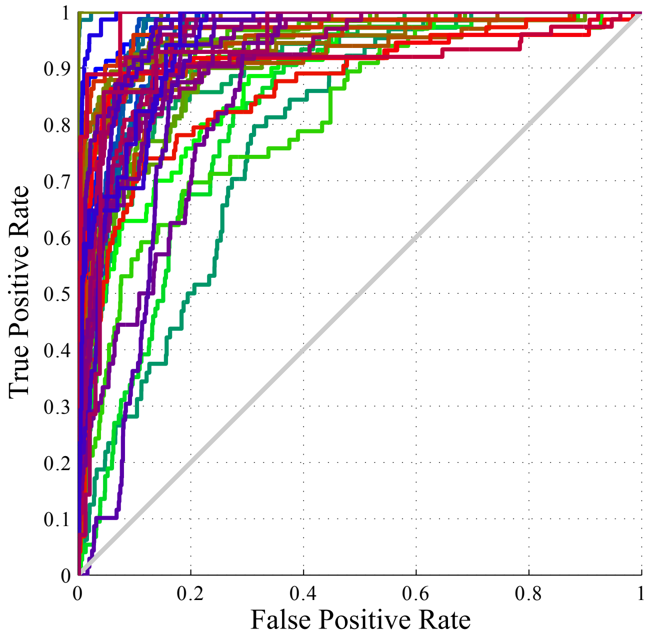


Fig. 2.8 – ROC curve for training (*Odia* samples).

Fig. 2.9 – Instances of look alike characters in *Odia* language.

Table 2.1 – Overall accuracy comparison for the OHCS 1.0 dataset.

Method	Rate of accuracy in (%)
Zernike moments [37]	72
Curvature feature [59]	69
F-ratio feature [60]	71.25
Skeletal convexity [61]	71
CFNC	80.25

2.3.3 Experiment on English and Bangla Samples

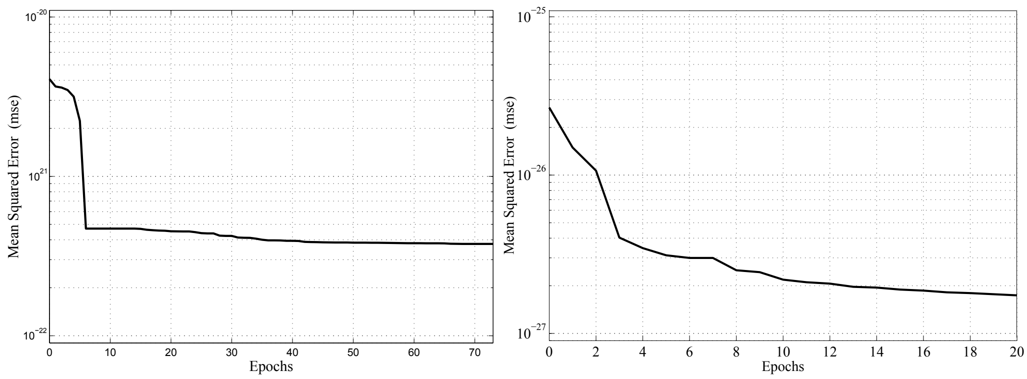
The proposed CFNC scheme is implemented on the character datasets of two more languages, namely, *English* and *Bangla*. The *English* database [57] consists of 1, 980 handwritten characters (26 upper-case letters and 10 decimal digits) with 55 samples from each. Similarly, the numeral dataset for *Bangla* language has 500 handwritten numerals with 50 samples each. Among the 1, 980 *English* samples, 1, 080 samples (30 from each class) are used for training and 900 (25 from each class) are used for testing. Among the *Bangla* dataset, 300 (30 from each class) samples are used for training and 200 samples (20 from each class) are used for testing. Training convergence characteristics so obtained are shown in Fig. 2.10(a) and Fig. 2.10(b) for *English* and *Bangla* datasets respectively. Both of these are at par good with their convergence properties. The ROC curves for both of these cases are also shown in Fig. 2.11 and Fig. 2.12. For (*English*), some curves corresponding to the characters 'F', 'M', 'Q', and 'W' are found to be less divergent towards the axis indicating a lower rate of accuracy. Samples of the characters 'M' and 'W' are found to be misclassified to each other with misclassification rates of 18% and 14% respectively. One of the main reason for this is the vertically reflexive structure of these characters. The overall rates of accuracy in both the cases are found to be 92% and 93.5% respectively which is quite satisfactory. The proposed CFNC scheme, implemented for all these three languages is compared with other competent features namely, Zernike moments, curvature feature, F-ratio feature, and skeletal convexity feature. The results are outlined in Table 2.3. In all the three cases, CFNC scheme is found to be efficient than the others. This shows the advantage of the scheme in classifying data among reasonable numbers of classes. A good rate of recognition for the three languages indicates the robustness of the CFNC scheme.

Table 2.2 – Rates of misclassification using CFNC scheme for homogeneously shaped characters.

Class	**Similar class**	**Misclassification rate(%)**
ଅ ('ah')	ଥ ('tha')	22
ଥ ('tha')	ଅ ('ah')	14.5
ଇ ('ih')	ଉ ('uh')	10.5
ଇ ('ih')	ଲ ('la')	8
ଇ ('ih')	ର ('ra')	6.5
ଉ ('uh')	ଭ ('bha')	19
ଭ ('bha')	ର ('ra')	10
ଷ ('sha')	କ୍ଷ ('kshya')	12.5
୨ ('two')	୭ ('seven')	11.5
୨ ('two')	୬ ('six')	9
୬ ('six')	୭ ('seven')	24
୭ ('seven')	୬ ('six')	15
୭ ('seven')	୨ ('two')	12

Table 2.3 – Comparison of classification accuracy of CFNC scheme with competent schemes.

Sample Type	**Feature**	**Rate of Classification (%)**	
		Train	Test
Odia	Zernike moments [37]	75	72
	Curvature feature [59]	74	69
	F-ratio feature [60]	74	71.25
	Skeletal convexity [61]	77	71
	CFNC scheme	**88**	**80.25**
English	Zernike moments	89.5	84
	Curvature feature	84.75	81.5
	F-ratio feature	90	82
	Skeletal convexity	84	76.75
	CFNC scheme	**92**	**90.75**
Bangla	Zernike moments	88	82
	Curvature feature	90.25	83.75
	F-ratio feature	85.5	80.5
	Skeletal convexity	91.5	86
	CFNC scheme	**93.5**	**87.5**



(a) Training convergence characteristic (*English* samples).

(b) Training convergence characteristic (*Bangla* samples).

Fig. 2.10 – Training convergence plots for the *English* and *Bangla* datasets.

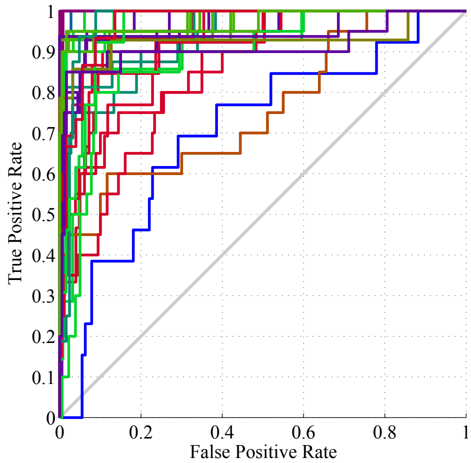


Fig. 2.11 – Training ROC (*English* samples).

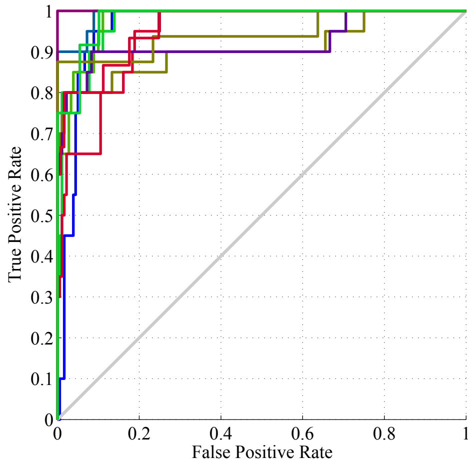


Fig. 2.12 – Training ROC (*Bangla* samples).

2.4 Summary

An efficient scheme (CFNC) for recognition of handwritten *Odia* characters is proposed in this chapter. Three primary features namely length, angle, and ratio are extracted from arc segments derived from the shape contour of a character. These features are used in a neural classifier to recognize the characters. Other existing features are also considered for recognition using the neural classifier to perform a comparative analysis. From the exhaustive simulation study it has been observed that the proposed CFNC scheme outperforms others with respect to the rate of recognition accuracy. The scheme has been extended for character recognition for the datasets of other languages namely, *English*, and *Bangla*. For these languages also, the CFNC scheme is found to be performing well in terms of recognition accuracy. However, the proposed CFNC scheme fails to perform accurately for characters of similar shape.

Chapter 3

Aggregated Features for *Odia* OCR using HMM Classification

In the previous chapter, we have suggested contour-based features for handwritten *Odia* character recognition. Each character is segmented into thirty segments with the centroid of the character as the base point. The features are namely, l_i (length from the centroid to starting point of a segment), θ_i (angle between l_i and chord joining i^{th} and $(i+1)^{th}$ segment), and r_i is the ratio between the i^{th} segment and its corresponding chord. The total number of segments being 30 for each character, each feature is of 30-dimensional and total length of the feature vector becomes 90 when concatenated together. The recognition has been studied on three different languages, *Odia*, *English*, and *Bangla*. In this chapter, an attempt has been made for feature reduction of the suggested contour features. We have utilized a fuzzy inference system (FIS) to generate an aggregated feature vector (AFV) for each character of length 30. This AFV has been used to classify the character using hidden Markov model (HMM). The final feature vector is divided into six levels based on its value and interpreted as six different states for HMM. Fifty-seven different six-state ergodic HMMs are thus constructed corresponding to fifty-seven distinct classes of characters and requisite parameters are calculated from each model. For the recognition of a probe character, its log-likelihood against these models are computed to decide its class label. The proposed scheme is implemented on the OHCS 1.0 *Odia* dataset and a recognition rate of 84.5% has been achieved. Finally, the scheme is compared with other competent schemes.

The chapter is organized as follows. Section 3.1 gives an overview on relevant works. Section 3.2 describes the proposed aggregated feature generation and HMM modeling used in the scheme. Section 3.3 outlines the experimental evaluation of the proposed scheme on handwritten *Odia* dataset. Section 3.4 summarizes the overall work proposed in this chapter.

3.1 Related Works

HMM has been reportedly used in [62] and [63] for recognition of handwritten numerals. A hybrid HMM along with ANN has been used, where the structural parts of the optical modes are modeled with Markov chains and ANN has been used for estimating the emission probability [63]. In [64], HMM has been used for statistical representation of the *Nushu* character strokes. Prior knowledge of character structures is used for the learning algorithm and likelihood scores of probes are estimated from a Gaussian Mixture Model (GMM). A sliding window technique has been implemented for printed *Arabic* character recognition [65]. Shapes of each of the 26 characters have been modeled uniquely. In [66], an aggregated part based method for handwritten digit recognition has been proposed. A key advantage of this method is its robustness against deformation. Investigations on a two-layered segmented HMM architecture has been used in [67]. Character models are defined as mixtures of allograph models with a variant shape of a character where autographs correspond to specific writing styles. HMM-based *Odia* numeral recognition using class conditional probability has been proposed in [68]. The HMM is generated out of the shape primitives of an individual numeral and is referred to as a template for establishing a match for probe numerals.

Employing ensembles of HMMs, investigations have been attempted in [69] for alphanumeric character recognition. The HMMs are generated using an incremental learning algorithm. A hybrid classifier based on HMM and fuzzy logic has been proposed in [70]. Fuzzy rules are applied to classify the HMM for each stroke into further sub-patterns based on the primary stroke shapes. Features are modeled primarily on statistics and structures of character shapes. Technically feasible description of the Delaunay triangle from the stroke segments has been justified in [71] for online recognition and extraction of handwritten characters. Here, triangularization is practiced to associate similar structures in specific groups. In [72], $n-$gram modeling is associated with HMM modeling for recognition of *Thai* and *English* characters and Multi-directional island-based projection has been utilized for feature representations.

It is observed from the literature that very few investigations on *Odia* handwritten characters have been reported so far. In this work, we have attempted to recognize *Odia* characters, numerals in particular using three primary features from shape contour of each character image.

3.2 Proposed Scheme

This section outlines the proposed AFV (aggregated feature vector) generation followed by HMM modeling for handwritten character recognition scheme. The underlying probabilistic structures in handwritten characters are not directly observable. Though, each class of character has to follow a particular geometric constraint, it can be modeled using HMM. This geometric structure can be represented in terms of a certain number of states and their state-transition probabilities. Further, the system's perception about an input character image is represented as random variables. The distribution of the variables is dependent on an observation of a particular state transition system. These observations can be considered as a sequence of values that represent the character image. The variation in a feature vector is generally modeled as a function of a single independent variable. This is quite suitable for speech recognition systems where, time is the natural choice as the single independent variable. However, in OCR, there are at least two independent variables as text images are represented in two dimensions. Fortunately, the proposed feature extraction and selection scheme can efficiently represent a 2-D textual image into a 1-D sequence vector. After a successful training task, fifty-seven models are generated corresponding to 57 *Odia* characters and are recorded in a model database. The recognition is performed to evaluate the trained models on the input test samples which do not belong to the training dataset.

The whole process involves various sub-steps like character acquisition and representation, pre-processing, feature extraction, feature selection, and recognition. This scheme mostly thrusts on feature extraction, selection, HMM modeling and recognition. Necessary prerequisite phases have been adopted using standard image processing techniques. The schematic block diagram of the proposed scheme is shown in Fig. 3.1. Since the scheme deals with contour-based aggregated feature with HMM-based recognition, it is suitably named as aggregated feature vector with HMM recognizer (AFHMMC).

3.2.1 Feature Aggregation using FIS

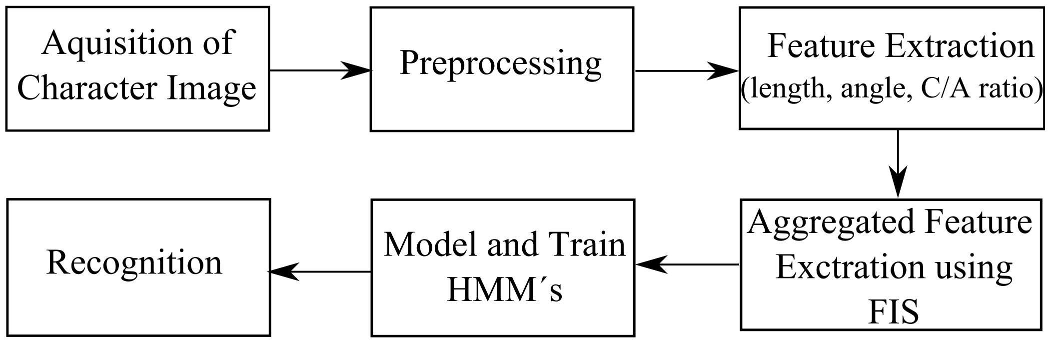


Fig. 3.1 – Overall block diagram of the proposed AFHMMC scheme.

In AFHMMC, each handwritten character is represented as an image of size 64×64. The shape centroid (C) is extracted, and the character shape contour is segmented into 30 different segments in a clock-wise direction in reference to the centroid. From each segment, three different primary features namely, length (l), angle (θ), and chord-to-arc-ratio (r) are extracted from various parameters of the segment. These three feature vectors from each character are fed to the next phase for aggregation using FIS. As seen from Fig. 3.2, there are four basic parts of a fuzzy inference model, namely, fuzzifier, inference system, rule-base, defuzzifier. Fuzzification is the first step in which the inputs are taken as crisp data. Values from the three primitive feature $(l,\ \theta,\ r)$ vectors are supplied as the input data. Input membership functions are quantified based on the degree to which the feature values belong to their respective set. We calculate the AFV for character recognition system by taking three input features to the FIS (Fig. 3.3). A set of twenty-seven **if-then** rules are utilized for the purpose. For defuzzifying the resultant of the output membership function, the 'centre-of-gravity' approach is followed.

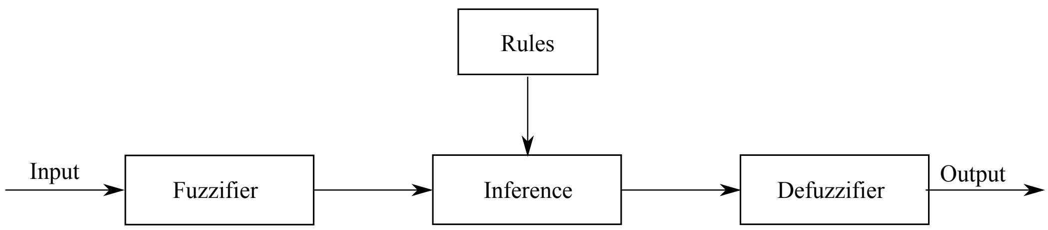


Fig. 3.2 – Components of FIS.

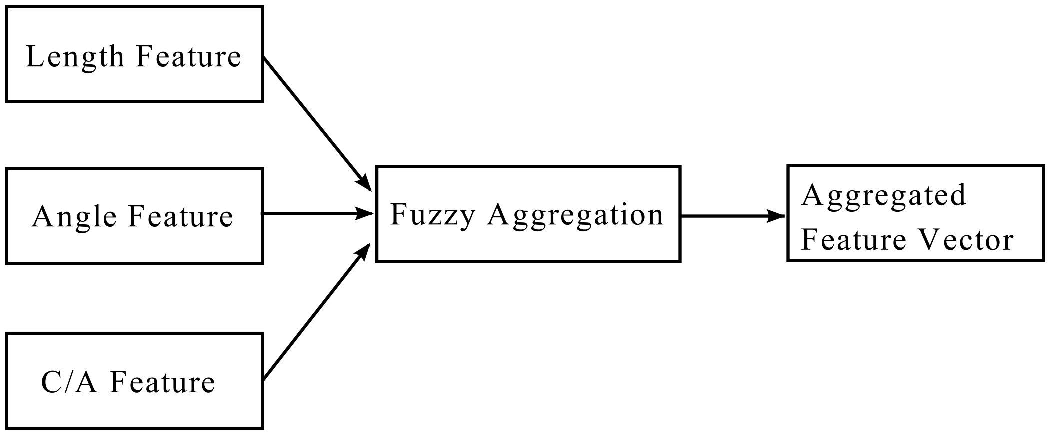


Fig. 3.3 – Contour feature input to FIS for feature reduction.

Since there are 30 segments per character image, we have 30 different lengths, angles, and chord-to-arc-ratios corresponding to each character. It may be observed, these three primitive features (l, θ, and r) for each character, create a feature space of dimension 1×90. Considering the dimension of feature space for each character, a feature reduction step is followed to generate an aggregated feature of size (1×30) from a (1×90) feature vector. The primary objective of using FIS for the feature aggregation process is to select the best feature value (among l, θ, and r) for each arc on the contour of a character. An aggregated feature vector (AFV) of size 1×30 is generated out of this fuzzy selection. The steps used to generate the reduced-dimension AFV is explained below in a sequel.

(a) **Fuzzification:**

During fuzzification, the feature points of each of the three primitive feature vectors are classified into three range of values namely, LOW, MED, and HIGH. The triangular membership functions are used for fuzzification. The membership functions used are shown in Fig. 3.4. The set of membership functions can be mathematically expressed as,

$$\mu_{LOW}(x) = \begin{cases} 1 - \frac{x}{p_b}, & 0 \leq x \leq p_b \\ 0, & otherwise \end{cases} \tag{3.1}$$

$$\mu_{MED}(x) = \begin{cases} \frac{x-p_a}{p_b-p_a}, & p_a \leq x \leq p_b \\ \frac{x-p_c}{p_b-p_c}, & p_b \leq x \leq p_c \\ 0, & otherwise \end{cases} \tag{3.2}$$

$$\mu_{HIGH}(x) = \begin{cases} \frac{x-p_b}{1-p_b}, & x \geq 1 \\ 0, & otherwise \end{cases} \tag{3.3}$$

The values of p_a, p_b, and p_c are experimentally set to be 0.25, 0.50, and 0.75 respectively.

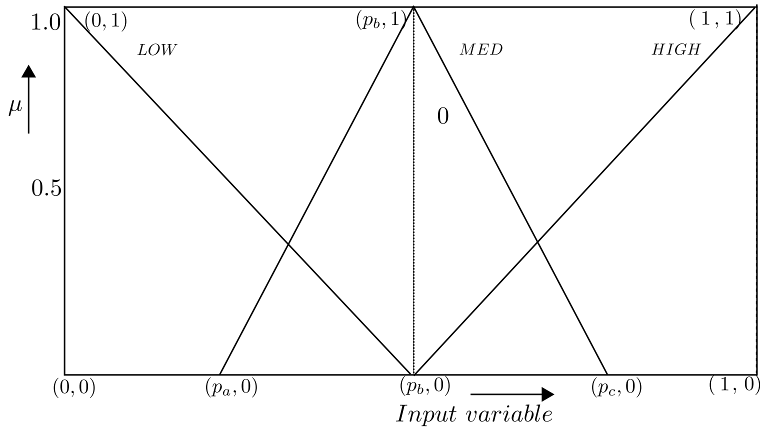


Fig. 3.4 – Triangular membership functions used in FIS.

(b) **Inference rules:**

For three different variables with three level membership functions a set of 27 inference rules are defined as shown in Fig. 3.5. Every possible combination of the three feature values are included in these rules. For example, if distance feature has value 'LOW' and angle feature has value 'LOW', and chord-to-arc ratio value is also 'LOW', then corresponding resultant output should be 'VERY HIGH'.

Low distance feature

Angle feature \ C/A ratio	LOW	MED	HIGH
LOW	VERY HIGH	HIGH	MEDIUM
MED	HIGH	MEDIUM	LOW
HIGH	HIGH	MEDIUM	LOW

Med distance feature

Angle feature \ C/A ratio	LOW	MED	HIGH
LOW	HIGH	MEDIUM	MEDIUM
MED	MEDIUM	MEDIUM	LOW
HIGH	MEDIUM	LOW	LOW

High distance feature

Angle feature \ C/A ratio	LOW	MED	HIGH
LOW	HIGH	MEDIUM	LOW
MED	MEDIUM	LOW	VERY LOW
HIGH	MEDIUM	VERY LOW	VERY LOW

Fig. 3.5 – Inference rules used for generating AFV.

(c) **Defuzzification:**

For defuzzification, the max-min rule is used to generate the crisp values as,

$$AFV_i = max \begin{bmatrix} min\{\mu_{LOW}(l_i), & \mu_{LOW}(\theta_i), & \mu_{LOW}(r_i)\}, \\ min\{\mu_{MED}(l_i), & \mu_{MED}(\theta_i), & \mu_{MED}(r_i)\}, \\ min\{\mu_{HIGH}(l_i), & \mu_{HIGH}(\theta_i), & \mu_{HIGH}(r_i)\} \end{bmatrix} \tag{3.4}$$

$$\text{where,} \quad i = 1, 2, \ldots, 30.$$

Thus, we get an aggregated feature vector (AFV) of size 1×30 from the original size of 1×90. A representative diagrams for this reduction process are shown in Fig. 3.6 and Fig. 3.7. The whole process of mapping the crisp input to output corresponding to the twenty-seven different rules is depicted in Fig. 3.8.

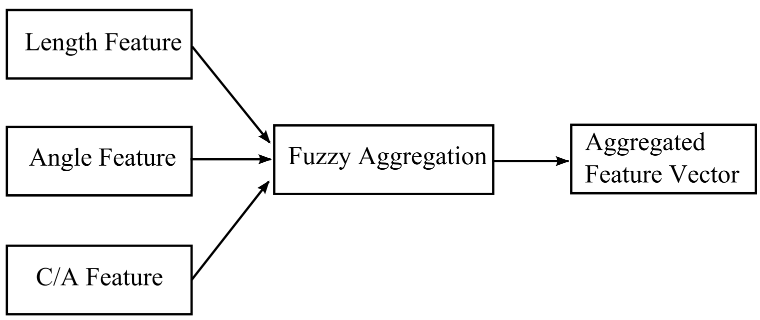


Fig. 3.6 – FIS for feature reduction.

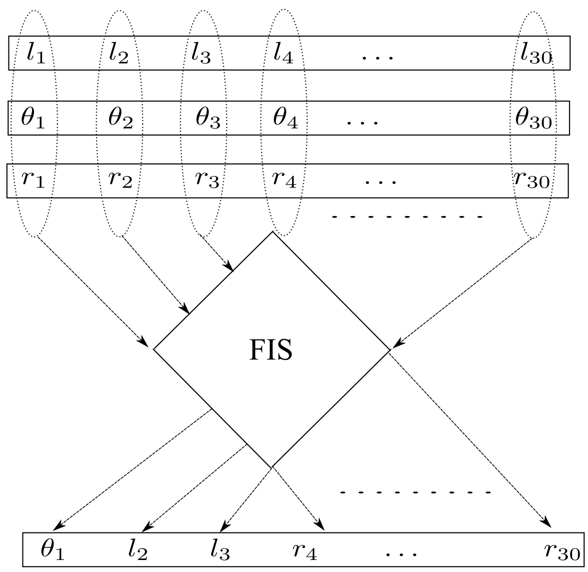
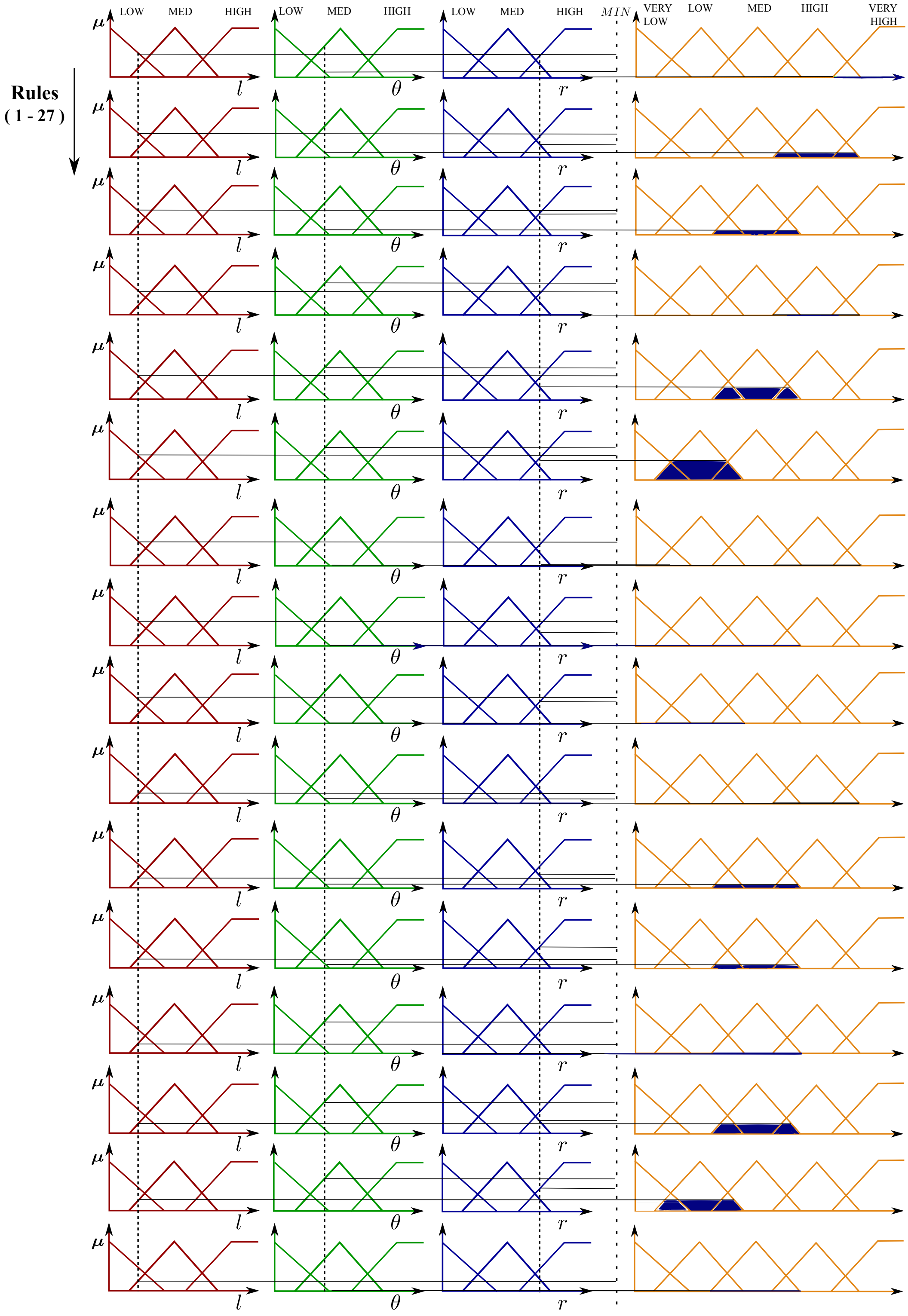


Fig. 3.7 – Feature reduction of 3×30 vector into 1×30 using FIS.


Rules
(1 - 27)
μ
LOW
MED
HIGH
l
θ
r
MIN
VERY LOW
LOW
MED
HIGH
VERY HIGH

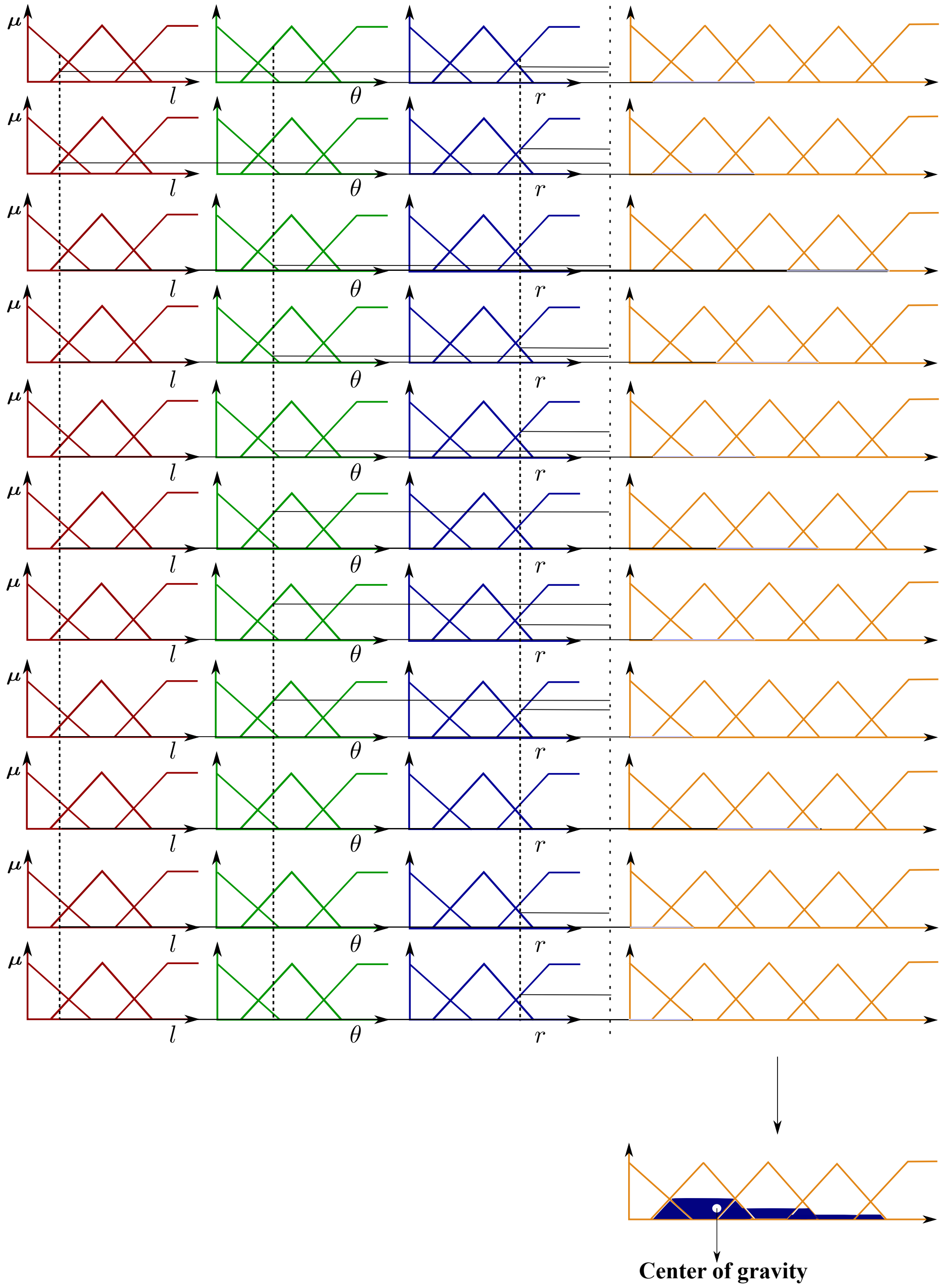


Fig. 3.8 – Membership functions for mapping of crisp input data to output.

3.2.2 Modeling the AFVs

HMM has vast applications in speech recognition systems like generating sequences of words in sentences, extraction of the sequence of phonemes in spoken words, etc. It is used to solve a large number of problems related to recognition [73]. HMMs are the statistical models in which the system, being modeled is considered as a Markov process that has unobserved or hidden states [74]. In hidden Markov models, the state is not directly visible, but only the sequence of observations influenced by the state at a particular instance of time are visible and hence the name Hidden Markov Models. A detailed illustration of HMMs is reported in [74], and [75]. A HMM can be mathematically define as,

HMM (termed as λ) constitutes a triplet (Π,A,B) where,

Π	=	(π_{ij}), the vector of initial state probabilities.
A	=	(a_{ij}), the state transition matrix.
a_{ij}	=	Pr $(x_{i_t}\|x_{j_{t-1}})$, probability of changing state from i to j, t and $t-1$ represents the present and previous instances respectively.
B	=	$(b_i(O_j))$, the confusion matrix or observation matrix.
$b_i(O_k)$	=	Probability of observing symbol O_k from the state x_j defined as $\Pr(O_k\|x_j)$

HMMs consist of two phase stochastic processes with the observation properties as hidden and visible. The first phase is a discrete Markov process that has a finite number of states and their transition probability values. Nature of this Markov process is only of *first order*, which means, the transition among two states is restricted and it relies only on the immediate predecessor state.

$$P(s_t|s_1, s_2, \ldots, s_{t-1}) = P(s_t|s_{t-1})) \tag{3.5}$$

The second phase in an HMM, produces a sequence of outputs (emissions) for every point temporally. For an instance, at any time point t, the output O_t is visible and the series of inhibit states s_t remains unknown. i.e.,

$$P(O_t|O_1, O_2, \ldots, O_{t-1}, s_1, s_2, \ldots, s_t) = P(O_t|s_t) \tag{3.6}$$

There are three primitive problems in HMM which need to be addressed and solved in order to use HMM effectively. These problems are dubbed as evaluation, decoding and optimization problems. A brief explanation about these primitives are listed below.

1. **Evaluation problem:** The evaluation problem deals with computing how well a given sample of observation matches with an existing model; i.e, it finds the probability of a observed sequence $(\Pr(O|\lambda))$ for an HMM (λ). This is

also called as the *recognition problem*. It evaluates which HMM produces the most probable observation sequence among the different existing HMMs for categorization among different classes.

2. **Decoding problem:** This is related to find the probable sequence of hidden states for a given set of observation sequence, i.e. $P(O|S)$. This is also called decoding of a recognition problem. Here it is necessary to find a maximum likelihood path of hidden states with a given observation sequence. The Viterbi algorithm is used for solving this type of problem.

3. **Optimization problem:** This is used for generating an HMM for a given sequence of observations, also known as the learning problem. It is the most difficult problem for recognition. Here the objective is to find the HMM parameters, i.e. $\lambda = (\Pi, \mathrm{A}, \mathrm{B})$. Each HMM is trained with the observed sequence to generate these parameters. Baum-Welch method is utilized for the purpose [76].

In the proposed AFHMMC scheme, six-state ergodic HMMs for each character are modeled. We have to train fifty-seven models corresponding to 57 *Odia* characters. The six hidden states are labeled in the range of $0-10$, $11-20$, $21-30$, $31-40$, $41-50$, and $51-60$. Similarly, the observation states are labeled in range of $0-5, 6-10, 11-15, 16-20, 21-25, \ldots\ 56-60$. These twelve observed levels for six hidden states (two observed sub-levels per hidden state) represent the participation of data points group wise. Six-state HMM is shown in Fig. 3.9. The range of values in hidden and observation states are chosen considering the aggregated feature vector (AFV) values. The initial probability matrix is of size 1×6, the state transition matrix of size 6×6, and the observation matrix is of size 6×12. These matrices are estimated using the forward-backward learning algorithm.

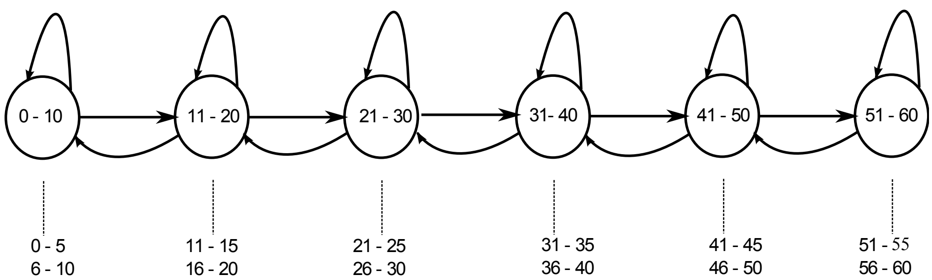


Fig. 3.9 – Six-states HMM.

The entries for the transition matrix A are computed by looking into the trend of two consecutive values of the AFVs, e.g. if a pair of consecutive data points are 5 and 18, then the state transition will be from state $0-10$ to the state $11-20$. For initializing the vector Π, equi-probable values $(0.17, 0.17, 0.17, 0.17, 0.16, 0.16)$ have been assigned to its six fields. The values for observation matrix B is calculated as,

The formulae used for computing various parameters are given as,

$$\alpha_1(i) = \pi_i b_i(O_1) \tag{3.7}$$

$$\alpha_{t+1}(i) = b_i(O_{t+1}) \sum_{j=1}^{n} (\alpha_t(j) a_{ji}); where, 1 \leq t < T \tag{3.8}$$

$$\beta_1(i) = \pi_i b_i(O_1) \tag{3.9}$$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(O_{t+1}) \tag{3.10}$$

$$\gamma_t(i) = \frac{\alpha_t(i).\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \tag{3.11}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \tag{3.12}$$

After reading the observation data from the first sequence, the entries in A, B and Π are updated using the Baum-Weltch algorithm (**Algorithm 3**) [74]. The training rules are executed as per the following update equations,

$$\Pi_i^{new} = \gamma_1(i) \tag{3.13}$$

$$a_{ij}^{new} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{j=1}^{N} \sum_{t=1}^{T-1} \xi_t(ij)} \tag{3.14}$$

$$b_i(v_k)^{new} = \frac{\sum_{t=1,O_t=v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{3.15}$$

where, α and β are forward and backward probability respectively, γ is the overall probability of being in state i at time t, ξ is the probability of transition from state i at time t to state j at time $t+1$.

Algorithm 3: Baum-Welch($dataset, \Pi, A, B$)

for *Each class of characters in the dataset* **do**
  1. Initialization:
  Consider a proper starting model $\lambda = (\Pi, A, B)$;
  2. Optimization:
  Update the model as, $\lambda' = (\Pi', A', B')$, such that:;
  $\Pi' = \gamma_1(i)$, $a'_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$, and $b'_j(O_k) = \frac{\sum_{t:O_t=O_k} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$;
  3. Termination:
  if $P(O/\lambda') = P(O/\lambda) + \epsilon$ **then**
    set, $\lambda = \lambda'$ and goto step-2;
  else
    Stop;
  end
end

In this training stage, fifty-seven different HMMs are designed corresponding to fifty-seven classes of handwritten characters. Similarly, for *English* characters and *Bangla* numerals, thirty-six HMMs, and ten HMMs are devised respectively.

3.3 Simulation Results and Analysis

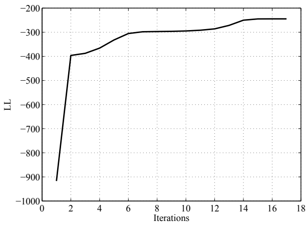
To validate the proposed AFHMMC scheme, simulation has been carried out in MATLAB 2011b environment for recognition of the handwritten *Odia* characters dataset. Primary features, l, θ, and r are extracted and subjected to feature reduction through FIS. Out of a total of $17,100$ samples, 30 samples from each character class are used for modeling each of the HMM, making the training sample size to be $1,710$ (10% of the total dataset). Some of the training convergence plots are shown in Fig. 3.10.

For testing, a total of $15,390$ samples containing 270 items from each character classes are considered. A probe character image is fed to the model test engine as shown in Fig. 3.11. Its maximum log-likelihood is captured against each HMM as,

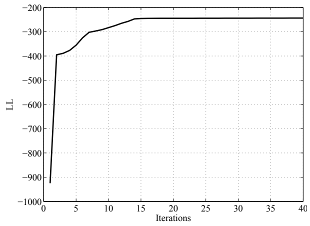
$$p(O|\lambda) = \sum_{i=1}^{n} \alpha_T(i) \tag{3.16}$$

The model test engine comprises of all the 57 trained HMMs. The log-likelihood ($LL_i; i = 0, 1, \ldots 57$) is computed for each model. The LL_i value of a probe is maximum for an appropriate class. A total of 15,390 character samples are tested using this test engine. Suitable modeling for the *English* and *Bangla* datasets are also carried out. For thirty-six different classes of the *English* dataset, 20 samples per class are considered with five states. Similarly, for ten different *Bangla* numerals, fifteen samples per class are used for training with three states.

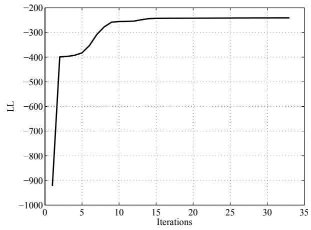
In AFHMMC scheme, recognition rates 84.5%, 92%, and 89.5% are achieved for the *Odia*, *English*, and *Bangla* datasets respectively. In Table 3.1, a comparison between the misclassification rates obtained for similar shape *Odia* characters using CFNC (Chapter 2) and AFHMMC is presented. There is a marginal increment of the rate of accuracy in almost all the cases of the look-alike characters. Along with the proposed scheme, other competent schemes are simulated using the same test dataset. The accuracy performance comparison for different schemes are shown in Table 3.2. It may be observed that the AFHMMC scheme shows improved performance over other competent schemes.



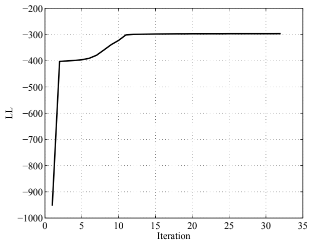
(a) LL plot of *Odia* character 'ah'



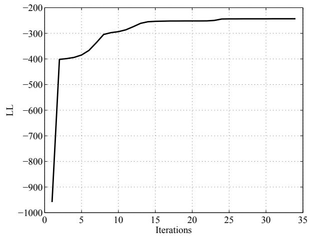
(b) LL plot of *Odia* character 'oh'



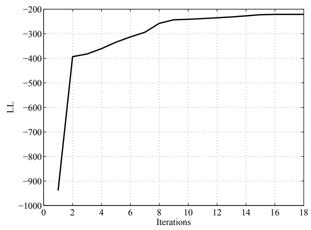
(c) LL plot of *Odia* character 'ka'



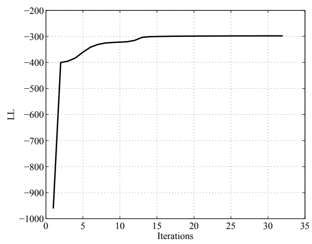
(d) LL plot of *Odia* character 'gha'



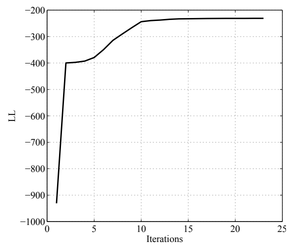
(e) LL plot of *Odia* character 'cha'



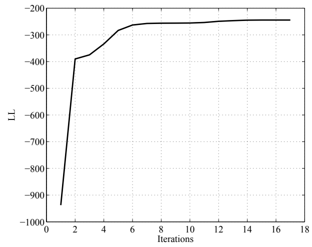
(f) LL plot of *Odia* character 'ma'



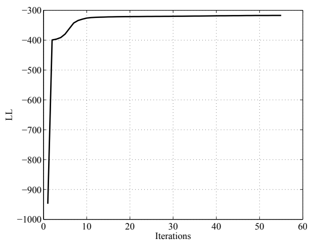
(g) LL plot of *Bangla* numeral '1'



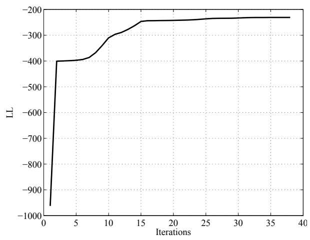
(h) LL plot of *Bangla* numeral '5'



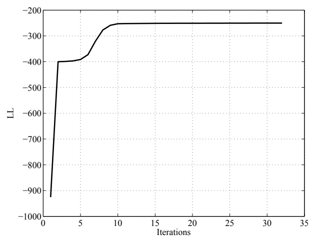
(i) LL plot of *Bangla* numeral '9'



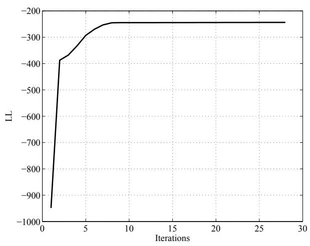
(j) LL plot of *English* character 'A'



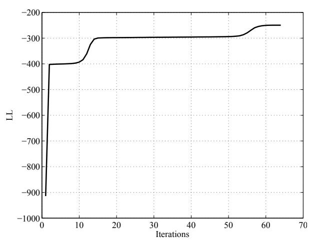
(k) LL plot of *English* character 'B'



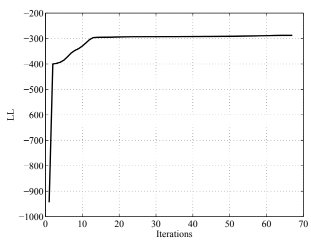
(l) LL plot of *English* character 'E'



(m) LL plot of *English* character 'H'



(n) LL plot of *English* character 'K'



(o) LL plot of *English* character 'T'

Fig. 3.10 – Plot of likelihoods versus iterations while training different handwritten characters.

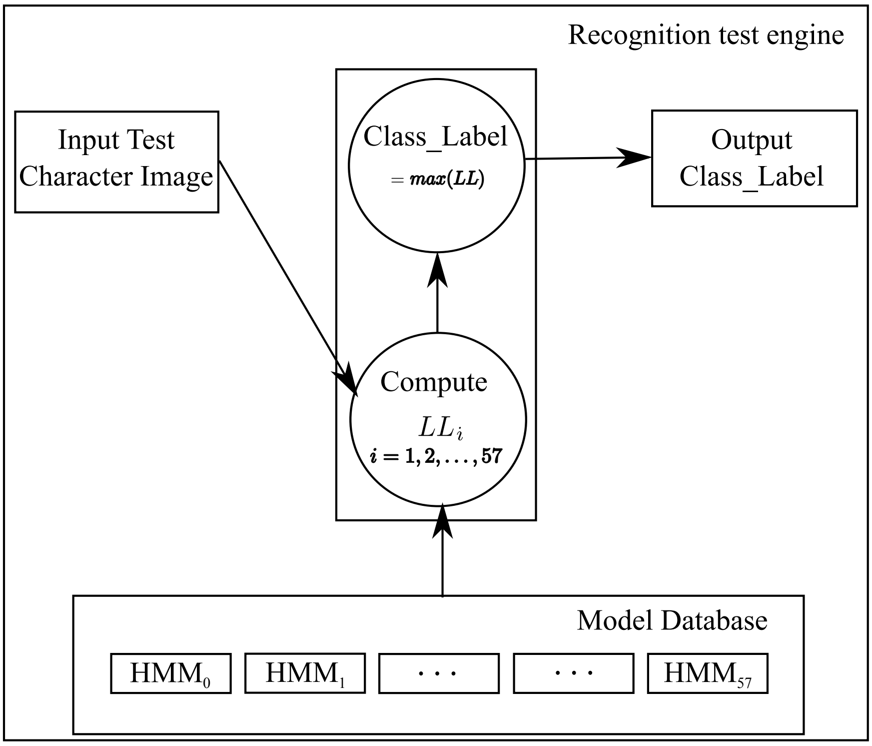


Fig. 3.11 – Model test engine for recognition using HMM.

3.4 Summary

This chapter deals with development of an HMM-based handwritten *Odia* character recognition scheme using aggregated feature vector. The scheme is abbreviated as AFHMMC. FIS is used for feature aggregation among the three primitive shape features (l, θ, and r) of a character. This process of aggregation helps in selecting appropriate feature among the three primitive feature for each segment of the shape contour of a character. This selection by aggregation yields the final feature vector of reduced length for a character image. HMM is used to train 57 different models for all the 57 classes of *Odia* characters. The proposed scheme is tested on the test samples of *Odia* dataset which gives an overall recognition rate of 84.5% . To set a justification for the use of feature aggregation using FIS. AFHMMC is compared with our CFNC scheme (**Chapter 2**). It is observed that the misclassification rate for look-alike characters is reduced in AFHMMC scheme as compared to the CFNC scheme.

Table 3.1 – Rates of misclassification using AFHMMC scheme compared with CFNC scheme for homogeneously shaped characters.

Class	**Similar class**	**Misclassification rate (%)**	
		CFNC scheme	AFHMMC scheme
ଅ ('ah')	ଥ ('tha')	22	21
ଥ ('tha')	ଅ ('ah')	14.5	12.5
ଇ ('ih')	ଉ ('uh')	10.5	10
ଇ ('ih')	ଲ ('la')	8	7.5
ଇ ('ih')	ର ('ra')	6.5	6.5
ଉ ('uh')	ଭ ('bha')	19	18
ଭ ('bha')	ର ('ra')	10	8
ଷ ('sha')	କ୍ଷ ('kshya')	12.5	10
୨ ('two')	୭ ('seven')	11.5	11
୨ ('two')	୬ ('six')	9	8.5
୬ ('six')	୭ ('seven')	24	22
୭ ('seven')	୬ ('six')	15	14
୭ ('seven')	୨ ('two')	12	11.5

Table 3.2 – Overall accuracy rate comparison of proposed AFHMMC scheme with other competent schemes.

Sample Type	**Feature**	**Rate of Classification (%)**	
		Train	Test
Odia	Sliding window + HMM[65]	79	75
	Part based + HMM [66]	74	68
	n-gram + HMM [72]	79.5	74.5
	CFNC scheme	88	80.25
	AFHMMC scheme	**88.75**	**84.5**
English	Sliding window + HMM	89.5	86
	Part based + HMM	84.75	86.5
	n-gram + HMM	90	80
	CFNC scheme	92	90.75
	AFHMMC scheme	**93**	**92**
Bangla	Sliding window + HMM	88	83
	Part based + HMM	90.25	85.25
	n-gram + HMM	85.5	82
	CFNC scheme	93.5	87.5
	AFHMMC scheme	**95**	**89.5**

Chapter 4

Development of an Evidence Collection based Local Feature for *Odia* OCR

Global shape features have been considered so far in Chapter 2 and 3 for the purpose of character recognition. These features, based on the shape contour of a character, have performed well in terms of the overall accuracy rate for 57 classes of *Odia* characters. However, a detailed investigation on the local shape features can not be ignored. In this chapter, our thrust is put on the local shape segments of the characters. In the previous propositions, traditional benchmark classifiers have been used for the purpose of recognition. However, there exists a scope to investigate for other classification schemes. In this context, a distance metric, namely, far_count has been proposed.

So far as the *Odia* characters are concerned, the uniqueness lies in one or more small parts of their structure instead of their whole structure. These parts of structure can be treated as evidence which can significantly represent each of their classes. In this work, such an evidence-dictionary has been prepared which is used subsequently for recognizing a test handwritten character. With this, the disturbance due to low profile characters and distorted characters can also be easily ignored.

Initially, the shape contour primitive (length parameter, l) of characters are used as input to our evidence collection algorithm. The proposed algorithm finds local meaningful segments out of these global patterns which are treated as evidence. The information gain is computed for every possible segment lengths of the contour primitives. The information gain for each local segment is found out by modeling a dedicated unique distance metric far_count. This is based on a splitting strategy among characters of 57 distinct classes with respect to that particular local segment. The local segment(s) with highest information gain values are finally accepted as evidence(s). Multiple number of such evidence may exist for a particular class of character (in case, the highest information gain values are identical for multiple sub-patterns). Thus, for all of these 57 characters, a dictionary of evidence segments is built. These evidence maximally represent their corresponding classes. Performance test carried out on a set of $14,250$ test samples from our dataset shows a quite satisfactory accuracy rate of 88%. Further comparative analysis with other related and competent schemes have been conducted, where the efficiency of the proposed work is found to be quite higher in terms of rate of accuracy and the units of time requirement. It may be noted that this is the first ever attempt made based on local pattern features for the recognition of the whole set of the handwritten *Odia* characters (all atomic alphabets and digits).

The rest of this chapter is organized as follows. In Section 4.1, an overview of the related works has been presented. Section 4.2 gives a detail explanation of the proposed evidence collection scheme. In Section 4.3, experimental evaluation of the proposed scheme on three datasets has been provided. A summary of the proposed evidence collection scheme is presented in Section 4.4.

4.1 Related Works

The recognition scheme proposed for one language is not necessarily a good recognizer for other languages. This is mainly due to the inter-variability among the structure of characters among different languages.

A local feature based script recognition scheme has been proposed for bilingual documents [77]. Experiments are carried out on *Tamil* and *English* language scripts present in the same document image. Font faces and sizes of printed characters are considered as basis for feature extraction. They suggest modeling of tetra bit values (TBV) per character, that corresponds to spatial spread. This spatial spread is induced from the segmented grids of individual character images. A reduced feature set based recognizer is proposed for unconstrained hand-printed symbols [78]. The proposed recognition scheme utilizes thinning algorithms to generate centre line thinned images from scanned symbols. Only those necessary topological, geometrical and local measurements which are needed to recognize the symbol are considered. Structured local edge pattern (SLEP) features are used in [79] for document text image classification. Their algorithm depicts an image's spatial distribution in the nearby edge direction. Four different samples namely: photo, slide, paper and table are considered for the experiment.

Discriminative local information based handwritten Chinese character recognition (SHCCR) is proposed in [80]. A manifold learning based subspace learning algorithm is introduced for the purpose. This brings discriminative locality alignment (DLA). Then after, the kernel version of DLA is introduced for kernel discriminative locality alignment (KDLA). KLDA is proved to be nothing but the subspace spanned by the DLA. A recognizer based on local energy feature and a local structural feature is proposed for isolated Arabic characters [81]. Convolution of skeletonized character image with log-Gabor filter bank is conducted to extract local energy features. Loops, end-points and other supporting dots in the skeletonized image are also used along this local feature in the final stage of the proposed recognizer. ANN is used as classifier for these transformation invariant features. A Persian character recognizer is presented for online OCR. Generic features of Persian characters are divided into local vectors using an LLNF (Local Linear Neuro-Fuzzy Model) technique [82]. A localized ellipse model is proposed for Chinese handwritten OCR. Refined sub-regions are extracted from user-defined neighborhood of specified pixel set in a character [83]. These sub-regions are integrated to form the intermediate feature vector. This intermediate feature vector is then projected into eigenspace using principal component analysis (PCA) to generate final feature vector. Support vector mechanism is used for classification purpose.

Incidentally, no recognizer has been suggested for *Odia* handwritten alphabets. Further, a majority of the state-of-the-art existing methodologies take into consideration the global features.

4.2 Evidence Collection

Evidences are those specific segments from the shape contour description of a character which are responsible for uniquely representing them. These segments can be used for the purpose of recognition instead of using the whole contour description. Such segments contain more information. To identify such local segments, we need to compute the information gain contents of all such possible segments that can be extracted from a whole contour descriptor of a character. However, another data structure, namely, 'trie' is utilized for generating such possible segments to reduce the computation cost. Among these segments, the one having highest information content is treated as the local feature evidence. The block diagram of the proposed scheme is shown in Fig. 4.1. Overall procedures of finding the local feature evidence are explained below in a sequel.

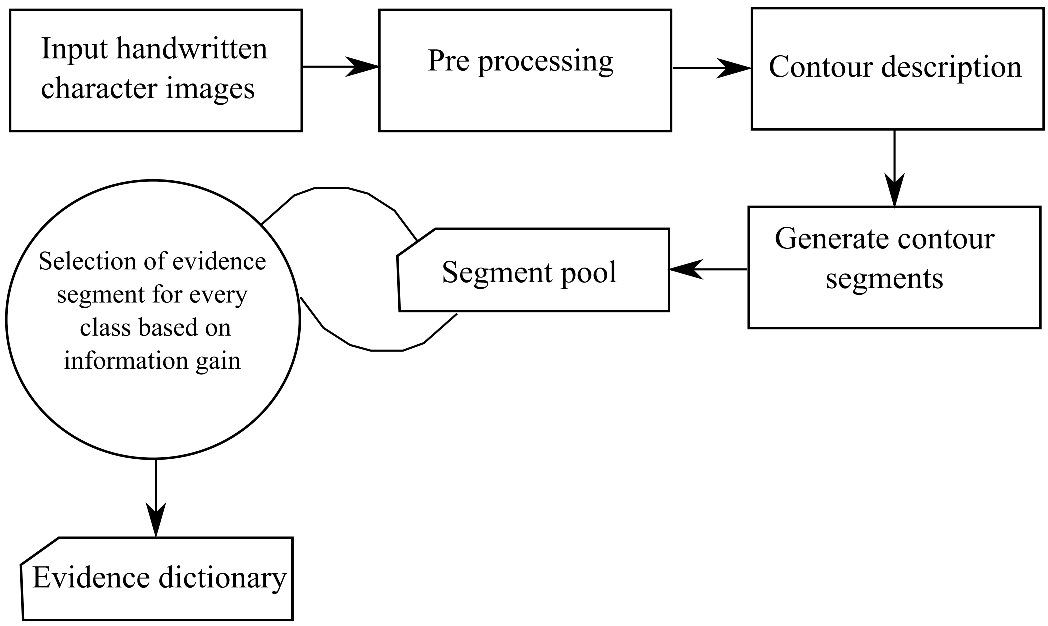


Fig. 4.1 – General overview of the proposed scheme.

4.2.1 Generating the Qualifiers Pool

A sliding window procedure (**Algorithm 4**) which runs with user-specified maximum (Max_{len}) and minimum (Min_{len}) range of lengths can be used to extract all segments. But, the computation cost involved in this process is too high. For example, suppose we need to extract all possible segments of lengths from all the character shape contour from a database. The number of such segments would be $avg \times size$, where, $avg = \frac{Max_{len}}{Min_{len}}$ and $size$ represents the total number of contour descriptors from the dataset. Searching for the segments with high information content would consume computation time of the order $O(avg^3 size^2)$. This is just an intractable procedure.

Thus, we follow a unique segmentation procedure (**Algorithm 5**) that efficiently utilizes the trie data structure for the purpose of segmentation. The trie data structure (also called as "prefix tree") is an ordered structure that is used to store a dynamic set of elements where the nodes are searched through prefixes. For reference, a diagrammatic representation of extraction of segments from the shape descriptor of the handwritten *Odia* character ଖ ('kha') is given in Fig. 4.2. The keys in a trie are conceptual tokens or strings. The steps followed to extract suitable segments from a character shape descriptor are as listed below.

1. The data points corresponding to shape descriptor of a character are inserted into a trie.

2. At each node, the frequency count is recorded for its key.

3. At each node, the entropy value for its key is computed. Finally, the entropy value for a particular node is computed based on the number of children it has. For example in the Fig. 4.2, the nodes with key values '3' and '4' in level-2 of the trie are having two children each. Thus, the entropy for these nodes are $(-\frac{1}{2}log_2\frac{1}{2}) - (\frac{1}{2}log_2\frac{1}{2})$. Similarly, the adjacent node with key value '5' will have entropy as $(-\frac{1}{3}log_2\frac{1}{3}) - (\frac{1}{3}log_2\frac{1}{3})$.

4. Voting expert method is followed with both of the above mentioned parameters (frequency and entropy) to put break points after the winning keys in the main descriptor. A sliding window of size 3 is run over the segment. For each run, if any of the frequency and entropy values of the three data points exceeds a specified threshold then only a break point is put after that data point.

5. Segments are extracted from the shape descriptor using break points.

The segments extracted using the above-mentioned steps are treated as the qualifiers. These qualifiers form the qualifiers pool. One or more evidence for each class of character are chosen from this pool. **Algorithm 5** outlines the steps for building the pool of qualifiers.

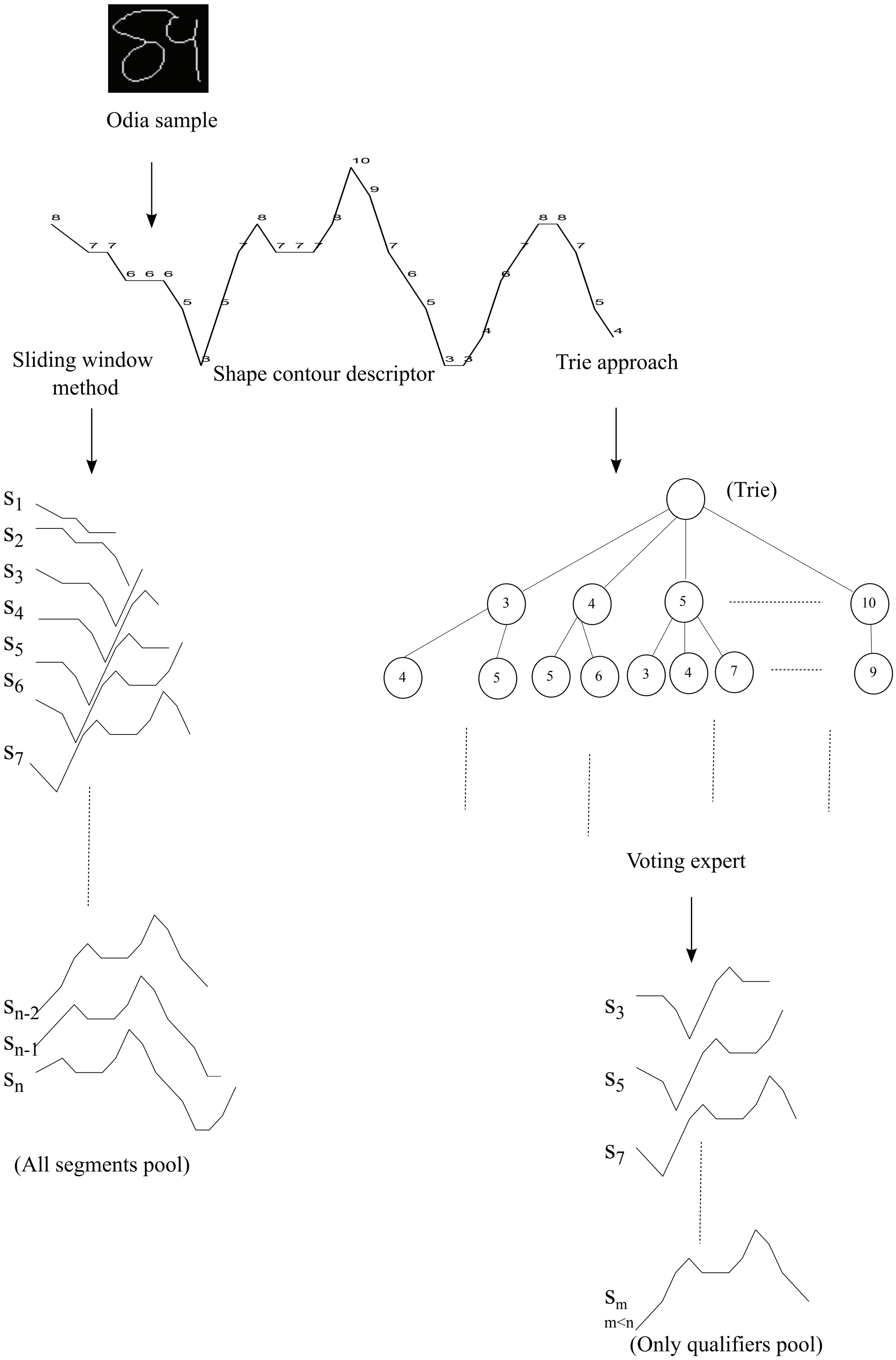


Fig. 4.2 – Representation of “sliding window” method and the trie-approach for extraction of segments from the shape descriptor of the *Odia* character ‘kha’.

Algorithm 4: Generate_Segments (Dataset, Max_{len}, Min_{len})

segment_pool $\leftarrow \phi$;
for *each* d_i *in Dataset* $i = 1, 2, \ldots, size$ **do**
    for $l = Max_{len}$ *to* Min_{len} *in step of -1* **do**
        l from d_i to *segment_pool*;
    end
end
Return *segment_pool*;

Algorithm 5: Generate_qualifiers(Dataset)

$trie \leftarrow \phi$, *segment_pool* $\leftarrow \phi$;
for d_i *in D,* $i = 1, 2, \ldots, size$ **do**
    Insert($trie$, d_i);
end
for *each* d_i *in trie* **do**
    if $vote(loc(d_i)) \geq threshold\_vote$ **then**
        qualifiers_pool $\leftarrow$ qualifiers_pool $\cup$ segments($loc(d_i), trie$);
    end
end
Return qualifiers_pool ;

Algorithm 6: FindEvidence (dataset D)

Qualifiers $\leftarrow$ GenerateSegments(D, Min_{len}, Max_{len});
best_gain $\leftarrow 0$;
for *Each q in* Qualifiers **do**
    gain $\leftarrow$ Evaluate_Qualifier(D,q);
    if gain > best_gain **then**
        best_gain $\leftarrow$ *gain*;
        best_Qualifier $\leftarrow q$;
    end
end
Return *best_Qualifier*;

Algorithm 7: Evaluate_Qualifier(Dataset, q)

for *each* d_i *in Dataset* `//` $i = 1, 2, \ldots, m$`;` m `is the number of qualifiers` **do**
    $Dist \leftarrow SegmentDist(d_i, q)$;
    insert q into number_line at par with $Dist$;
end
Return ComputeGain(number_line);

Algorithm 8: ComputeGain(number_line)

break_dist$\leftarrow$ Best_breakpoint(number_line);
Initialize D_1 and D_2 to ϕ;
for *key in number_line* **do**
    if *key.dist* $<$ *break_dist* **then**
        Allocate *key.character* to D_1;
    end
    Allocate *key.character* to D_2;
end
Return $H(D) - H'(D)$, (Eq. 4.2);

4.2.2 Computation of Information Gain for the Qualifiers:

Entropy is the measure of the degree of *purity/impurity* in a group of examples. It comes from the concept of information theory. It is the most common way to measure the degree of *purity/impurity* in any mathematical domain. The following simple equation outlines the expression for computing entropy in a group of objects,

$$entropy = \sum_i -P_i \log_2 P_i \tag{4.1}$$

where, P_i is the probability of class-i, it is computed as the number of objects belonging to class i present in the whole set.

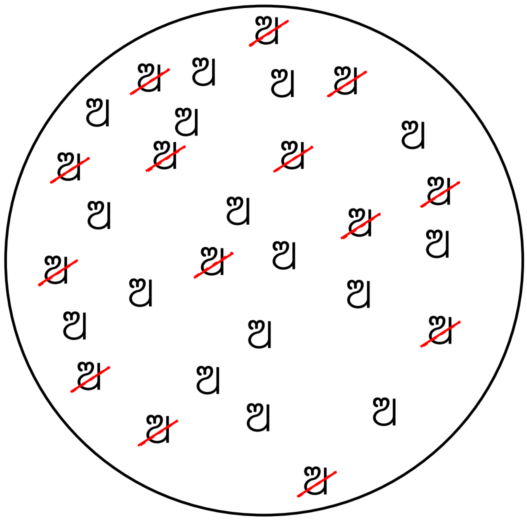


Fig. 4.3 – A 2-class example set for entropy calculation.

For simplicity, let's consider Fig. 4.3 that represents a set containing objects of two distinct classes namely, class-ଅ (class-1), and class-~~ଅ~~(not class-1). There are 16 objects of class-1 and 14 objects of class-2. In this set of 30 objects, the entropy can be computed as,
$entropy = -\frac{16}{30}\log_2(\frac{16}{30}) - \frac{14}{30}\log_2(\frac{14}{30}) = 0.99$

Information gain (I_g) is a measure of the classification ability of a specific pattern. It calculates how well a pattern can recognize its corresponding object distinctly in a set. Information gain is used for decision making (in decision trees) and classification. The working formula for information gain is: $I_g = entropy(Parent) - Avg\_entropy(Children)$. Mathematically it can be expressed as,

$$I_g = H(D) - H'(D) \quad or, \tag{4.2}$$

$$I_g = H(D) - [f(D_1).H(D_1) + f(D_2).H(D_2)] \tag{4.3}$$

where, $f(D_1)$ and$f(D_2)$ are the fractions of objects in subsets D_1 and D_2 respectively in the set D. $H(D)$ and $H'(D)$ are the entropy values before and after splitting the set D respectively.

Let's split the set as given in Fig. 4.3 into two subsets as shown in Fig. 4.4 where the classes may not be exactly partitioned.

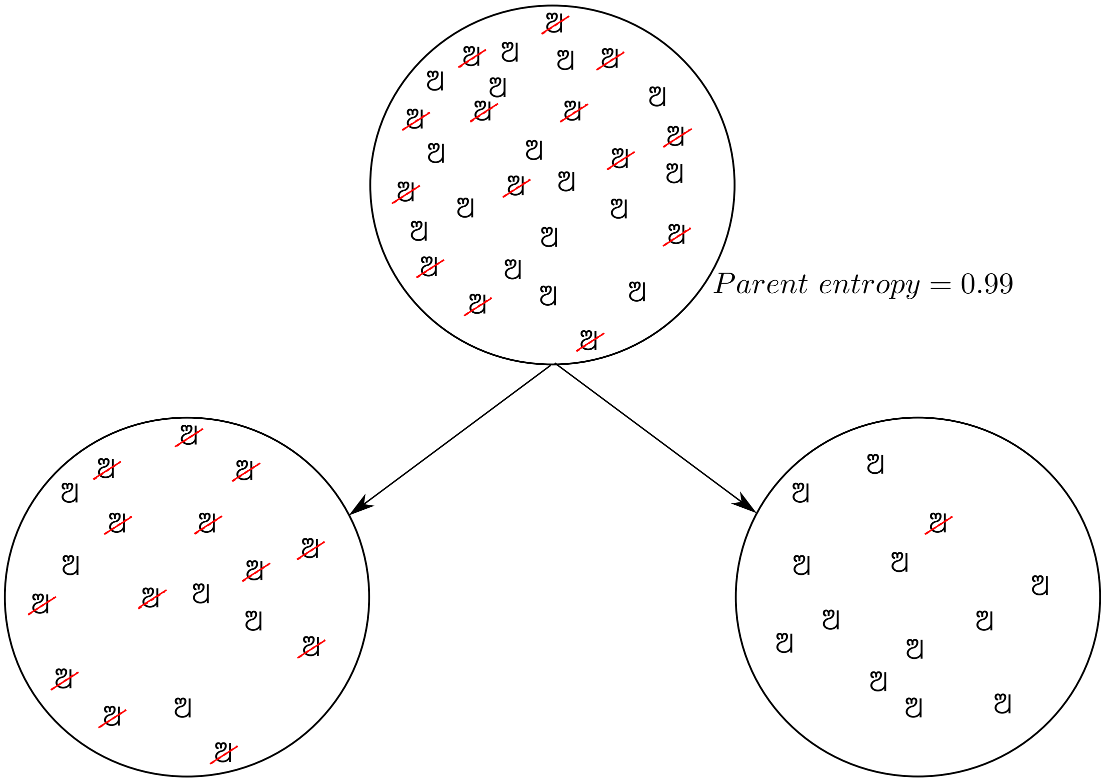


Fig. 4.4 – Splitting the set using a specific strategy.

For this, we can calculate the information gain for this particular split strategy as, $I_g = 0.99 - (\frac{17}{30} \times 0.787 + \frac{13}{30} \times 0.391) = 0.38$
This value represents the efficiency of how much this particular split strategy is for splitting the parent set into its children sets. Iterative steps may be followed to compute this information gain in case of sets containing objects of more than two classes.

Computation of far_count **:** For a particular shape descriptor (say, d_k), first, the distance values of q to all such segments from d_k having a length equal to that of q are found out. Here, all such segment refers to all possible segments and not only the trie-generated qualifiers. The Euclidean distance is a good metric for this purpose. However, the beneficial effect of classifying long segments is not taken into account by the Euclidean distance. Therefore, a dedicated distance measure is explored and modeled, dubbed far_count. It is the number of counts between any two segments of equal lengths which returns the number of corresponding data points among those segments which are farther to each other by a threshold. This threshold

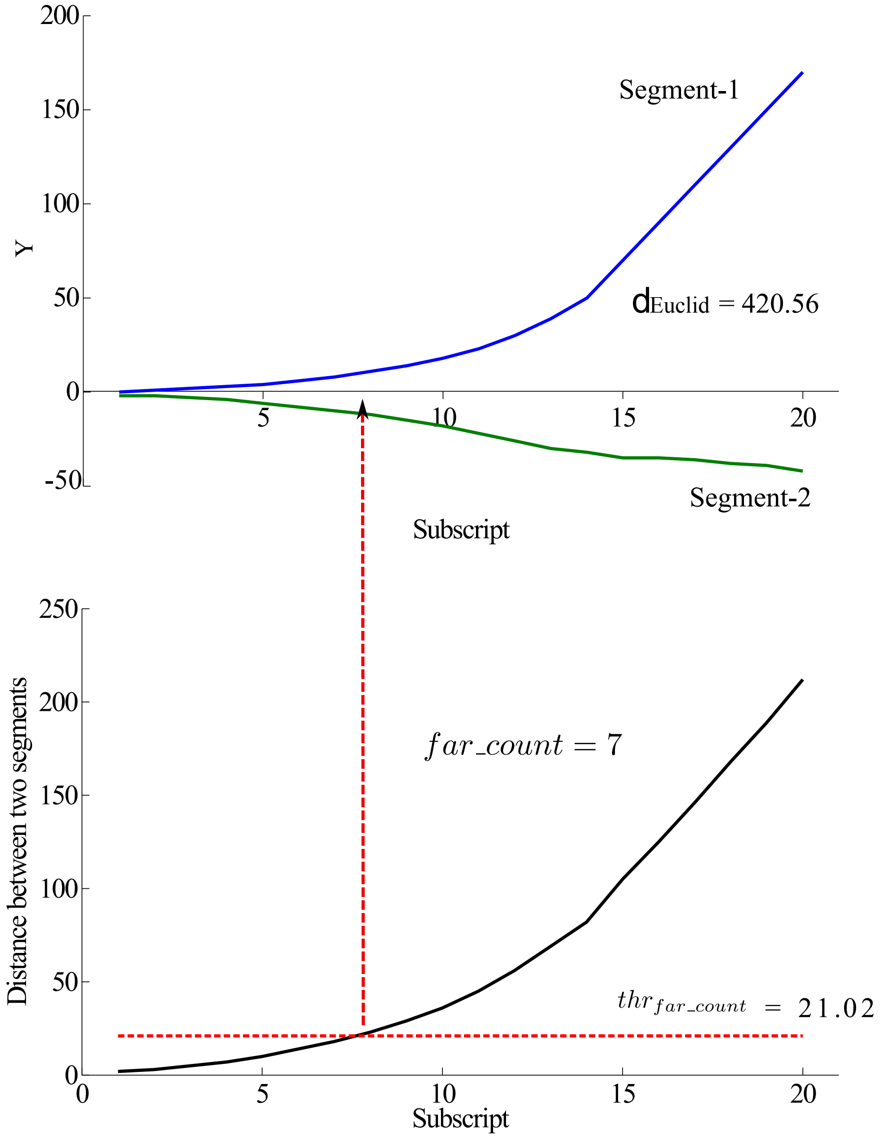


Fig. 4.5 – Schematic representation for far_count. For simplicity, only 'Y' is presented.

($threshold_{(Fc)}$) can be a heuristic choice. For our case, we define this threshold as the ratio of the overall Euclidean distance between two segments to the total number of data points. Mathematically,

$$thr_{(Fc)} = \beta.d_{Euclid(S_1,S_2)} \tag{4.4}$$

where, β equals 1, divided by the number of data points in each segment yielding a penalty on short segments. It is shown in Fig. 4.5 that two segments belong to a same class of character are having a high value of the Euclidean distance between corresponding points in two segments as compared to the far_count value. As, we are interested in computing the information gain value, utmost care is taken about the distance value between two segments. Because, lower the distance (far_count) value, the closer the segments are, and more is the corresponding information gain value. Thus, the risk of misclassifying an object of one class to another class is reduced. Further, for handwritten characters, the intra-class variations are minimized by the use of far_count.

The set of steps listed below are followed to compute the information gain (I_g) of a particular qualifier (q). It may be noted that computation of information gain is possible in case of a binary classification. The *Odia* character set consists of fifty-seven classes and for every character the whole set is partitioned into two classes, *i.e.* one belonging to one class and rest characters as another class. This process is repeated till each set belongs to separate class. The detail steps are listed below.

Step-1: Compute the distance from q to all the shape descriptors in the set.

Step-2: Set the distance between q and d_k as the lowest far_count between q and all segments from d_k which are of equal length to q. Calculate the far_count of q with all descriptors.

Step-3: Arrange the descriptors along a number line based on their respective far_count values. A representation is shown in Fig. 4.6.

Step-4 Consider a partition distance and partition the total objects into two sets. For example, in Fig. 4.6, there are 18 characters and for partition distance $= 2$, characters on the left are taken as set_1 and to the right are considered as set_2.

Step-5: Compute the information gain for the *qualifier* segment q using (4.2). In the current example (Fig. 4.6), the information gain for the qualifier q is computed as,

$$\begin{aligned} I_{g(q)} &= H(set) - H(set') \\ &= 0.99 - f(set_1).I_{g(set_1)} + f(set_2).I_{g(set_2)} \\ &= 0.99 - (\tfrac{9}{18}).((-\tfrac{6}{9})\log_2(\tfrac{6}{9}) - (\tfrac{3}{9})\log_2(\tfrac{3}{9})) + (\tfrac{9}{18}).((-\tfrac{2}{9})\log_2(\tfrac{2}{9}) - (\tfrac{7}{9})\log_2(\tfrac{7}{9})) \\ &= 0.27 \end{aligned}$$

where $H(set)$ is entropy of set before partition and $H(set')$ is for after partition.

Step-6: Repeat the above steps for all the qualifiers belonging to a specific class of character. The qualifier yielding highest information gain value is now treated as the evidence for that particular class. This qualifier along with the particular partition distance (for which giving highest gain value) is recorded into the evidence dictionary. It may be noted that there may exist multiple such qualifiers for a particular class of character.

Step-7: Repeat the above steps for finding evidence for all the class of characters and store them into the evidence dictionary.

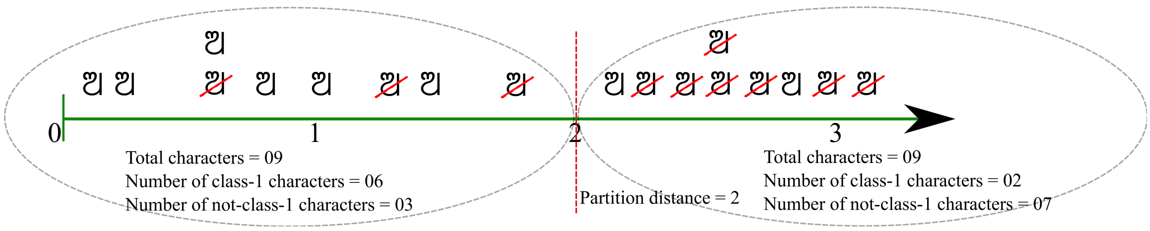


Fig. 4.6 – Arrangement of objects in number line based on their far_count to q.

The pseudo-codes for finding an evidence is illustrated in **Algorithm 6** along with its related sub procedures in **Algorithm 5**, **7** and **8**. These algorithms are specified for a binary classification for simplicity. Lets consider a binary dataset D with its objects labeled A or B as their class labels. **Algorithm 6** first generates all suitable (but not all possible) segments called as qualifiers using an trie data structure. Next, it checks the efficiency of each qualifier by evaluating its information gain value *i.e.*, how much it is closer in discriminating class-A objects from class-B objects. If this information gain value is higher than the priorly initialized bsf_{gain} value, then the bsf_{gain} value is updated with this higher value of information gain. At the end, the qualifier(s) with best information gain values is/are considered as the evidence(s) for their respective classes. These are recorded into the evidence dictionary along with their respective distance and gain values for future purpose of recognition. As, there are fifty-seven classes, the matching problem concerns a $one - versus - all$ comparison. In the training stage, we classify characters of class-1 against characters of class-2 to class-57. From this, the evidence segment for class-1 is stored in the evidence dictionary for this train set. Now, for remaining 56 classes the process is repeated *i.e.*, discriminating class-2 against class-{{1, 2, . . . , 57}-{3}}. Parsing, in the same way, the final evidence dictionary is built. However, there are some special cases, where the characters are disjoint in their structure and for such cases, a multiple numbers of evidence segments are recorded. Furthermore, here we can observe that **Algorithm 4** is a time and space consuming procedure and it is quite a brute-force.

A modified procedure has been suggested in **Algorithm 5** which incorporates the trie that has been successfully used for meaningful segmentation of data sequences.

The proposed scheme is evaluated on handwritten samples of three different languages (*Odia*, *English*, and *Bangla*). Training phase incorporates use of the far_count measure in place of Euclidean distance metric. To justify this replacement, a set of pilot experiments are carried out for the three languages. For the comparison purpose, three distance metrics are considered namely, far_count, Euclidean distance, and correlation measure. A k-fold ($k = 10$) cross-validation check is performed upon their training samples. The plot of results thus obtained are shown in Fig. 4.7. It is found that, apparently, using the far_count value in place of the Euclidean distance is giving better accuracy in all the three languages.

4.3 Experimental Evaluation

Among the $17,100$ *Odia* characters from the dataset, $5,700$ (57×100 samples per character) characters are used for training and rest $11,400$ are used for testing. Every character images are pre-processed, whereby, size standardization (64×64), binarization and thickening operations are carried out. Training is performed as per the **Algorithm 6**. As the first attempt, 100 samples from class-1 are considered along with 100 randomly selected (but at least one from each class) samples from the rest of the samples from class-2 to class-57. The output so obtained gives the evidence shape contour segment(s) from class-1 along with corresponding gain value and the threshold distance value. In the second attempt, the discrimination process is carried out for 100 samples from class-2 against 100 samples from class$\{\{1, 2, \ldots, 57\}$-$\{2\}\}$. The trend of gain values computed for every qualifiers for this instance is illustrated in Fig. 4.8. In the subsequent attempts, same procedure is followed for finding the evidence segments for all the classes of characters. The set of evidence thus collected now constitutes the evidence dictionary. The dictionary has basically two attributes, one is the class-wise evidence and the other is corresponding threshold distance value. Table 4.1 shows an instance of the evidence dictionary for *Odia* language. The information gain obtained for each evidence is mentioned against their description. Higher gain value for an evidence indicates its higher capability of discrimination.

For testing the correctness of the evidence finding algorithm, remaining 200 samples from each class are considered as probes. Thus, making the test sample set size to be $11,400$. A straight forward k-fold cross validation (k = 10) is applied.

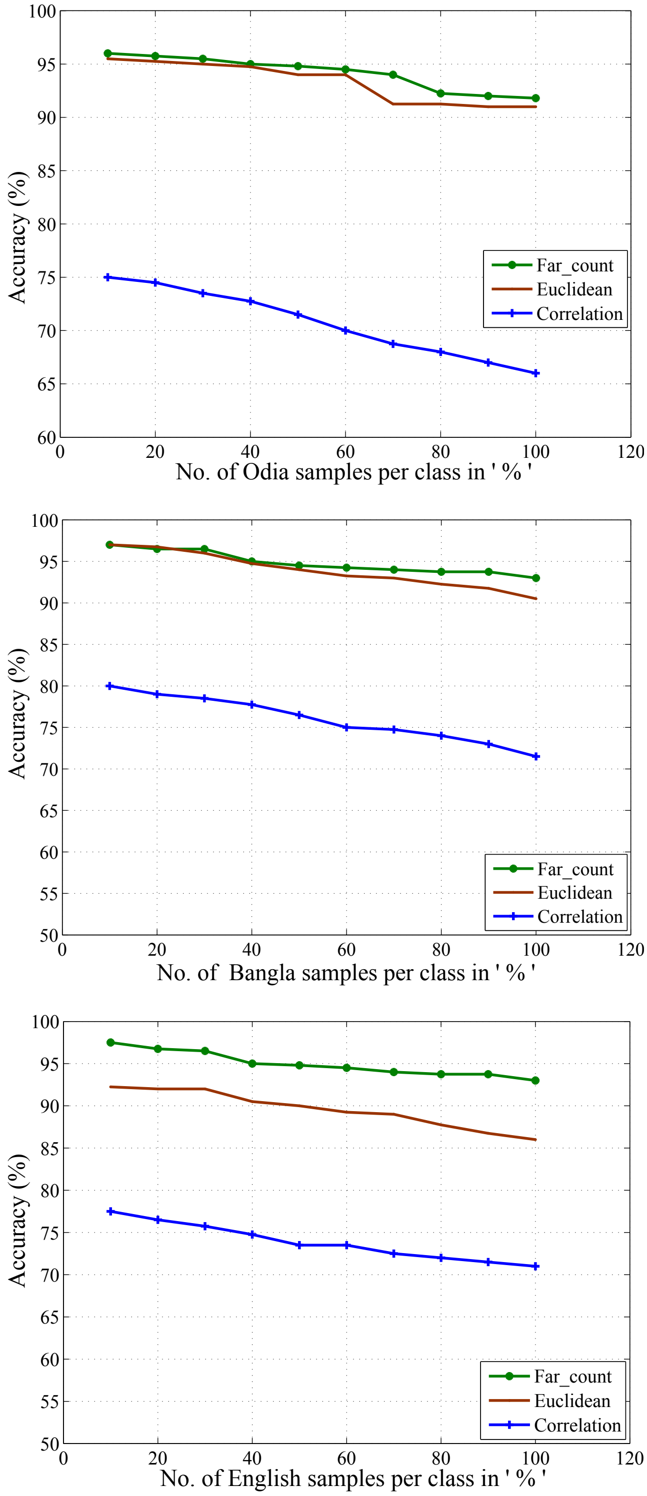


Fig. 4.7 – Rates of accuracy for different numbers of training examples for three distance metrics. The low performance of the correlation measure shows that an appropriate segment match is not sufficient to obtain satisfactory performance.

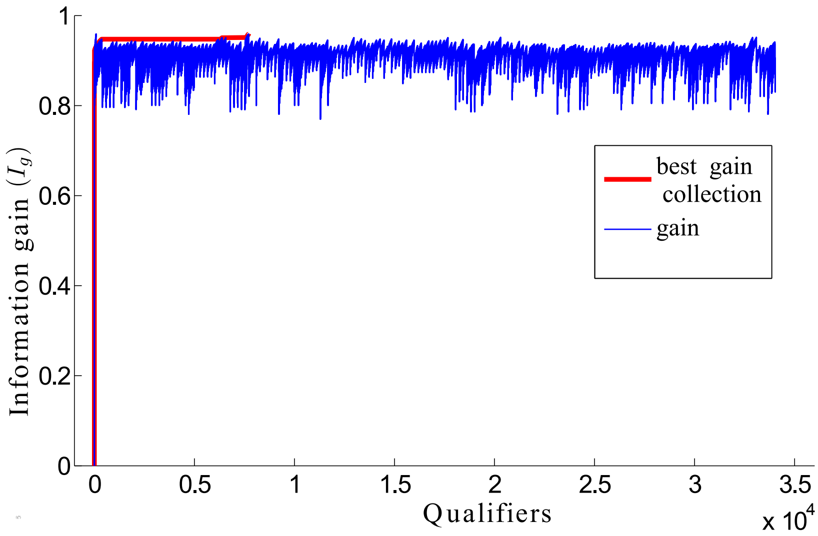


Fig. 4.8 – Plot of gain values of the qualifiers (as per **Algorithm 6**) with simultaneous collection of best gain values.

First, the contour descriptor of a test sample is found out after applying necessary pre-processing steps on it. Then, every evidence from the evidence dictionary are compared against the test sample contour descriptor by finding the distance from a descriptor to the evidence segment ($SegmentDist(D, S)$).

Finally, the evidence having least value of its distance to the test sample descriptor is selected, and class label of this evidence is assigned to that probe. Initially, a test sample set containing 10% from each class is randomly selected from the dataset and the overall rate of accuracy is found out. This experiment was repeated for 10 times and finally average rate of accuracy is computed. Subsequently, the volume size is then increased by 10%, and the same process is repeated until the complete dataset is tested.

The proposed scheme is tested for two other languages namely, *English* and *Bangla*. Handwritten characters of 36 classes (26 alphabets and 10 digits) from the *English* language and handwritten numerals of 10 classes from *Bangla* language are considered for this purpose. A total of 3,600 *English* samples are considered from which 1,800/1,800 samples are used as train/test samples. Similarly, a total of 500 *Bangla* samples are considered from which 300/200 samples are used as train/test samples.

The overall accuracy of 88% is achieved for the handwritten *Odia* dataset which is quite satisfactory for a reasonable 57 classes of characters. In Table 4.3, the misclassification rate of similar shape characters has been compared with that of the CFNC scheme (Chapter 2) and AFHMMC scheme (Chapter 3). Satisfactory improvement is apparently observable from this table. The local informative contour segments are easily able to distinguish between homogeneous shape characters which are far to reach using the whole shape descriptors. Similarly, overall accuracy rates of 94% and 90.5% are obtained for the *English* and *Bangla* samples respectively. The performance of the proposed scheme is compared with three other competent schemes. The results are outlined in Table 4.3. It is observed that the proposed scheme outperforms the other schemes for all the three languages. This shows the advantage of the scheme in classifying data among reasonable numbers of classes.

4.4 Summary

An efficient local shape descriptor based recognition scheme is thus proposed. A new distance metric far_count has been introduced. The scheme efficiently generates a evidence dictionary which contains meaningful local segments of the shape contour of characters. These evidence can be used for recognizing the class label of an isolated probe character. The scale and rotation invariance properties of these segments make the scheme more robust. Proper test has been conducted to evaluate the efficiency of the scheme. Accuracy rate of 88%, 95%, and 92%, have been obtained on the *Odia*, *English*, and *Bangla* test samples. Comparative analysis with other competent schemes shows that the proposed evidence collection is superior to others in terms of accuracy and misclassification rate.

Table 4.1 – Samples from the evidence dictionary

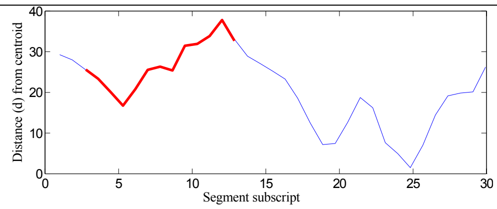
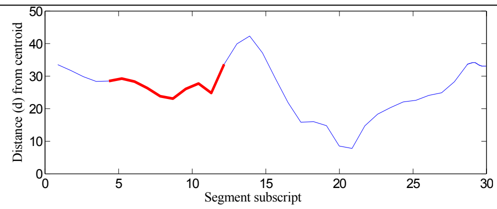
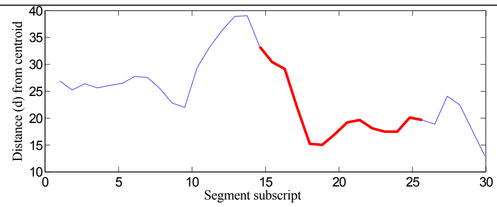
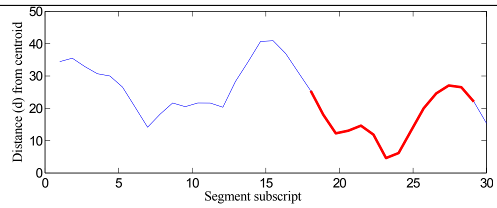
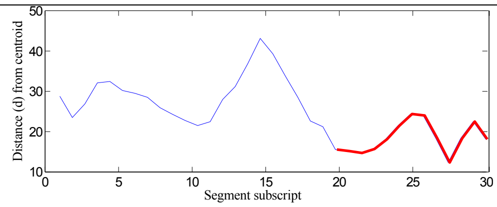
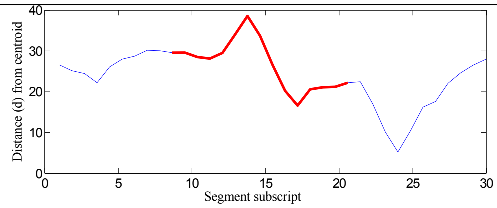
Character Class	**Descriptor**	**Gain value**
		0.96
		0.98
		0.98
		0.96
		0.90
		0.95

Table 4.2 – Improvement in the rates of misclassification using ECLF-FC for homogeneously shaped characters.

Class	Similar class	Misclassification rate (%)		
		CFNC	AFHMMC	ECLF-FC
ଅ ('ah')	ଥ ('tha')	22	21	16.5
ଥ ('tha')	ଅ ('ah')	14.5	12.5	11
ଇ ('ih')	ଉ ('uh')	10.5	10	6.5
ଇ ('ih')	ଲ ('la')	8	7.5	6
ଇ ('ih')	ର ('ra')	6.5	6.5	4
ଉ ('uh')	ଭ ('bha')	19	18	16
ଭ ('bha')	ର ('ra')	10	8	7.5
ଷ ('sha')	କ୍ଷ ('kshya')	12.5	10	6.5
୨ ('two')	୭ ('seven')	11.5	11	9
୨ ('two')	୬ ('six')	9	8.5	6.5
୬ ('six')	୭ ('seven')	24	22	15
୭ ('seven')	୬ ('six')	15	14	12
୭ ('seven')	୨ ('two')	12	11.5	10

Table 4.3 – Overall accuracy comparison of the ECLF-FC scheme with competent schemes.

Sample Type	**Feature**	**Rate of Classification (%)**	
		Train	Test
Odia	TBV feature + HMM[77]	74.25	72
	SLEP + ANN[79]	78	75
	KLDA scheme[80]	82	79.5
	CFNC scheme	88	80.25
	AFHMMC scheme	88.75	84.5
	ECLF-ED scheme	*90.5*	*86*
	ECLF-FC scheme	**92**	**88**
English	TBV	92.25	89
	SLEP	93.5	91
	KLDA	94	92
	CFNC scheme	92	90.75
	AFHMMC scheme	93	92
	ECLF-ED scheme	*93.5*	*92*
	ECLF-FC scheme	**96.25**	**95**
Bangla	TBV	87.75	86
	SLEP	90	87
	KLDA	93.5	91.5
	CFNC scheme	89	87.5
	AFHMMC scheme	95	89.5
	ECLF-ED scheme	*92*	*90*
	ECLF-FC scheme	**93.5**	**92**

Chapter 5

Hybrid Energy Feature based Handwritten *Odia* OCR

Image transformation algorithms are quite helpful for feature extraction and ultimate object recognition in both still and dynamic scenes. Transformations like DFT, DCT, DWT have been widely used in different pattern recognition schemes to derive features. Character recognition falls into the similar direction of research. These transformations can be fruitfully utilized to derive meaningful features for recognition. In this chapter, an attempt has been made to suggest an energy based feature using the DCT and DWT in combination. BPNN has been utilized as a classifier. Simulations have been carried out on datasets of three different languages namely, *Odia*, *English*, and *Bangla*. The performance analysis with respect to accuracy has been made with competent schemes. It is observed that the suggested hybrid feature has an upper hand in recognition accuracy.

The rest of the chapter is organized as follows. Section 5.1 outlines the related work on character recognition using transform-based features. Section 5.2 provides a detailed description on the development of the proposed hybrid energy feature. Experimental evaluations are presented in Section 5.3. Section 5.4 outlines the summary.

5.1 Related Works

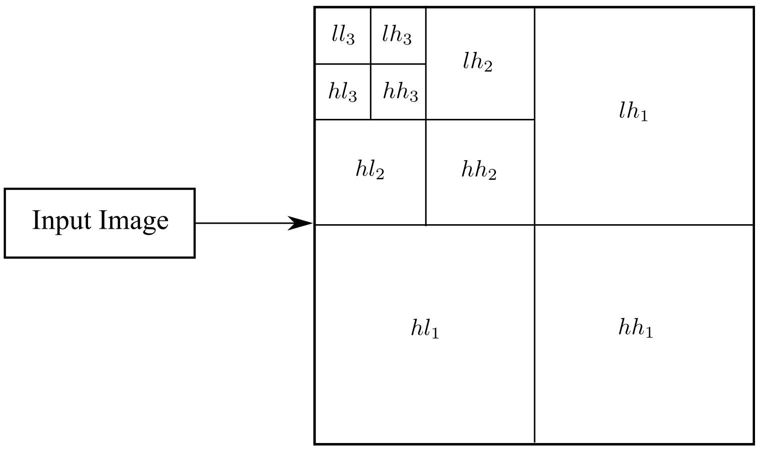
Degraded *Chinese* license plate recognition is achieved using wavelet transformation and fractal dimension [84]. This method pretends to work for blurred character images as well using shifting differential box-counting approach (SDBC). It is used to calculate the number of fractal boxes for each character image and generates feature vector. The wavelet transformation is also used for machine printed atomic *Chinese* character recognition [85]. A faster algorithm based on wavelet transformation is proposed for *Chinese* character recognition [86]. It transforms the binarized character image into one-dimensional (1-D) data along vertical and horizontal directions. For any character image of size $W \times H$, it suggests only $W + H$ operations, thus reducing the computation time. Further, the 1-D data is decomposed using Haar wavelet, and the information is extracted from low-frequency components.

A transformation technique based on partial inclination (TPID) is proposed for handwritten *Japanese* character recognition [87]. This technique basically aims at dissolving the distortions found in the characters. The algorithm constructs certain functions based on the angle of inclination of character images. These functions are used for the transformation to produce necessary features for recognition along with the correction of vertical and horizontal corrections. Feature extraction techniques based on wavelet, curvelet and ridge-let transformations are proposed for the multi-lingual character recognition [88]. The curvelet transformation based features are found to be good so far as the multi-lingual document image is concerned. Wavelet packet transformation is used for *Chinese* handwriting recognition [89]. Character images are first decomposed and their partial decomposition tree is chosen based on their variance property of energy function. The image parts are split into a number of local windows, and their respective energy values are concatenated to generate feature vector. PCA is then used for some feature point reduction. A noise-tolerant and affine-invariant gray scale handwritten *English* numeral character recognizer is proposed [90]. This recognizer implements the global affine transformation (GAT). It computes affine-invariant correlation of the input character image with the target character image. The topographic features are used as matching constraints.

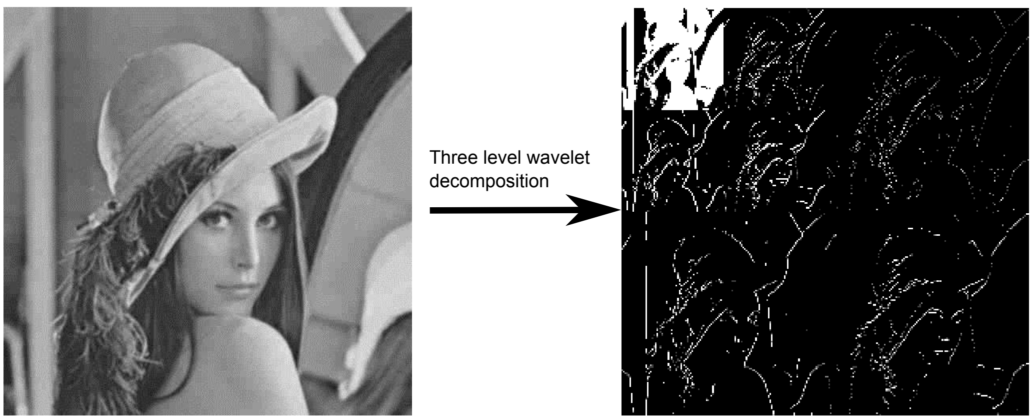
Image transformation algorithms have been proved to be efficient for the purpose of character recognition. Recognizers based on these techniques have been proposed for several Asian languages and volumetrically for the *Chinese* language. It has been observed that for *Odia* character recognition, no such scheme has been proposed using any transformation based features.

5.2 Proposed Feature using DWT and DCT

The Discrete Cosine Transformation (DCT) has a strong energy compaction property for which it has been widely used in various image processing applications. Since its proposition by Ahmad et al. [91], it has undergone repeated modifications [92–94]. Recently, a CORDIC based fast radix-2 DCT algorithm has been proposed which is appropriate for signals of even lengths [94]. Again, the signal decomposition approach considered here is simple enough and does not take care of a major issue, i.e. signal noise ignorance.



(a)



(b)

Fig. 5.1 – (a) Schematic diagram of wavelet decomposition. (b) Wavelet transform of Lena image up to three levels.

In this work, we have proposed a hybrid feature extracted from DWT and DCT in combination to recognize handwritten characters. Handwritten *Odia*, *English*, and *Bangla* characters have been taken for case studies. Simulation results have been analyzed to show the efficacy of the proposed feature extraction technique. The idea is to develop an energy based feature by combining both DCT and DWT algorithms to obtain a better recognition rate than the individual algorithms. The implementation and the experimental results of this work reflect the combined outcome of the DCT and DWT algorithms merged into a single hybrid algorithm.

The DWT is a more specific case of the family of wavelet transforms that retains both temporal and spatial information of an input image. The transform essentially divides the image based on the frequency components along both the horizontal and vertical directions by using high pass and low pass filters. Different wavelet families exist such as biorthogonal [95], Daubechies [96], discrete meyer [97] and reverse biorthogonal [98, 99]. Each of these consist of many types of wavelets such as Haar, Daubechies family(db2, db3), bior (biorthogonal family), rbio1.1 (reverse biorthogonal) and dMey (discrete Meyer family). The choice of a wavelet depends on many factors such as the task at hand as well as the compatibility of the given wavelet to other algorithms associated. A schematic three level decomposition by the DWT and three level decomposition of Lena image are shown in Fig. 5.1(a) and 5.1(b) respectively.

The DCT is widely used for lossy image compression that expresses the points of information as a sequence of cosine functions at different frequencies and amplitudes. The DCT compresses the energy of an image into a small number of coefficients. Maximum energy is located at the top left corner of the coefficient matrix known as DC-coefficient and other prominent energy coefficients are oriented in a zig-zag manner starting from the DC-coefficient. DCT coefficients, $f(u, v)$ of an input image $f(m, n)$ is given by (5.1), [55]

$$f(u,v) = \alpha(u)\alpha(v)\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)cos[\tfrac{(2m+1)\pi u}{2M}] \times cos[\tfrac{(2n+1)\pi v}{2N}] \tag{5.1}$$

Considering the properties of the DCT and DWT, the present scheme extracts feature from the characters employing the DWT and DCT in sequence. The input character image is subjected to the DWT and energy coefficients are extracted using DCT on detail sub-bands, i.e hh, hl, lh. If there are k levels of decomposition for each character, we get $3k$ number of detail sub-band images, labeled as $hl_1, lh_1, hh_1, hl_2, \ldots hh_{k-1}$. Since DC-coefficient represents the approximation and does not contribute the detailed features, we do not consider it for feature extraction. In fact, coefficients in all others contribute to feature. If we aggregate all the features, the dimension of the feature set will be very large. To reduce the dimension of the feature set, only energy coefficients are generated applying DCT to each sub-band and potential coefficients are considered to create the feature vector. More precisely, coefficients from m such images are concatenated to represent the overall feature vector of a character image. The set of steps involved in the proposed scheme is depicted in **Algorithm 9** The overall block diagram of the proposed scheme is shown in Fig. 5.2. Selection of coefficients and construction of final feature vector from a character image is shown in Fig. 5.3.

Algorithm 9: Hybrid_Energy_Feature_Extraction

```
for each character image in the dataset do
    Represent each handwritten character image as a gray image of size
    256 × 256;
    Use gray scale morphology for preprocessing the character image;
    Apply k-level DWT decomposition to generate 3k detailed sub-band images
    excluding the approximation image;
    Compute DCT of these images for dimensionality reduction of the DWT;
    Select m DCT coefficients from each sub-band image and concatenate them
    in a linear sequence to generate a feature vector (f_v) of size 3 * k * m;
end
```

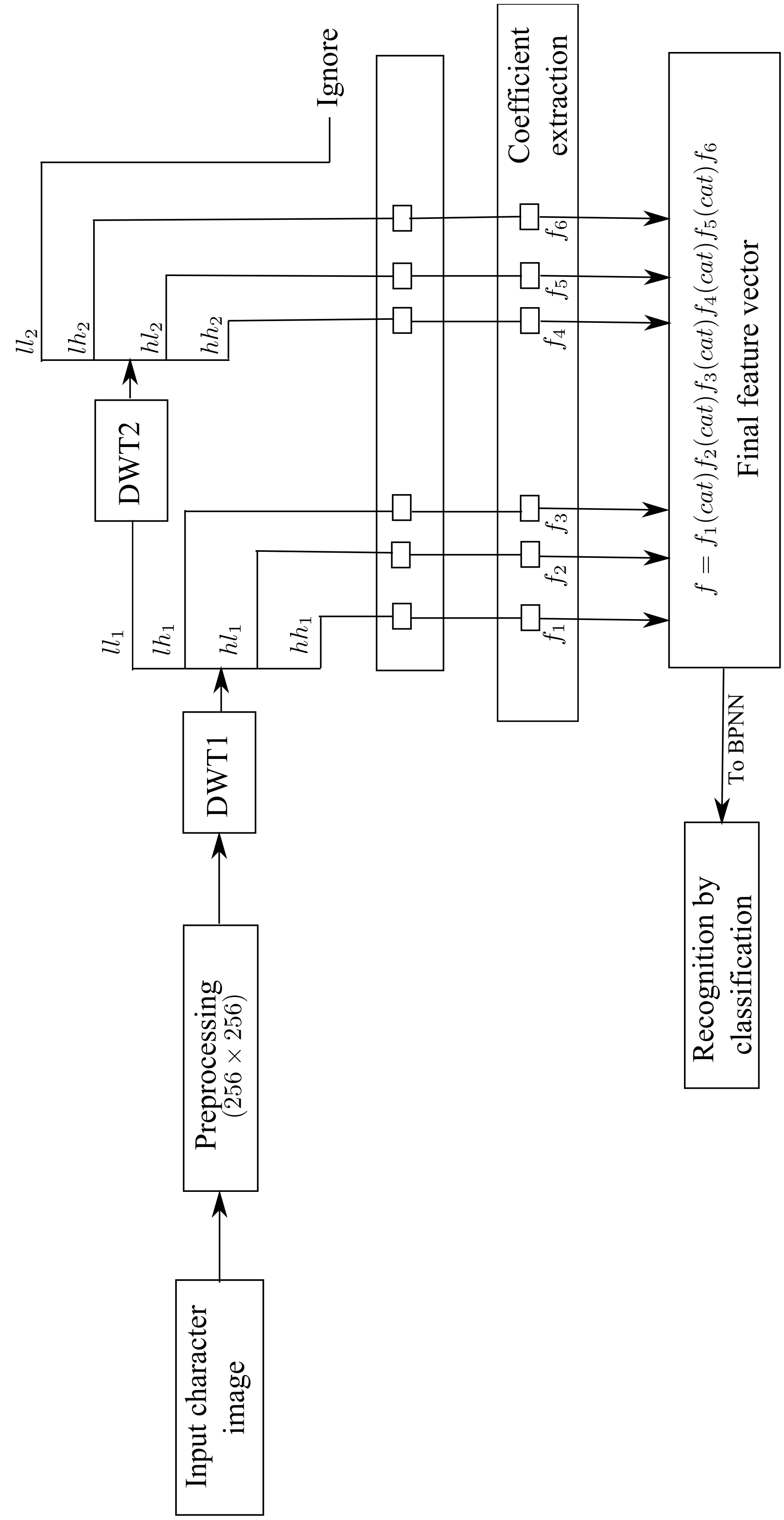


Fig. 5.2 – Block diagram of the hybrid feature extraction scheme.

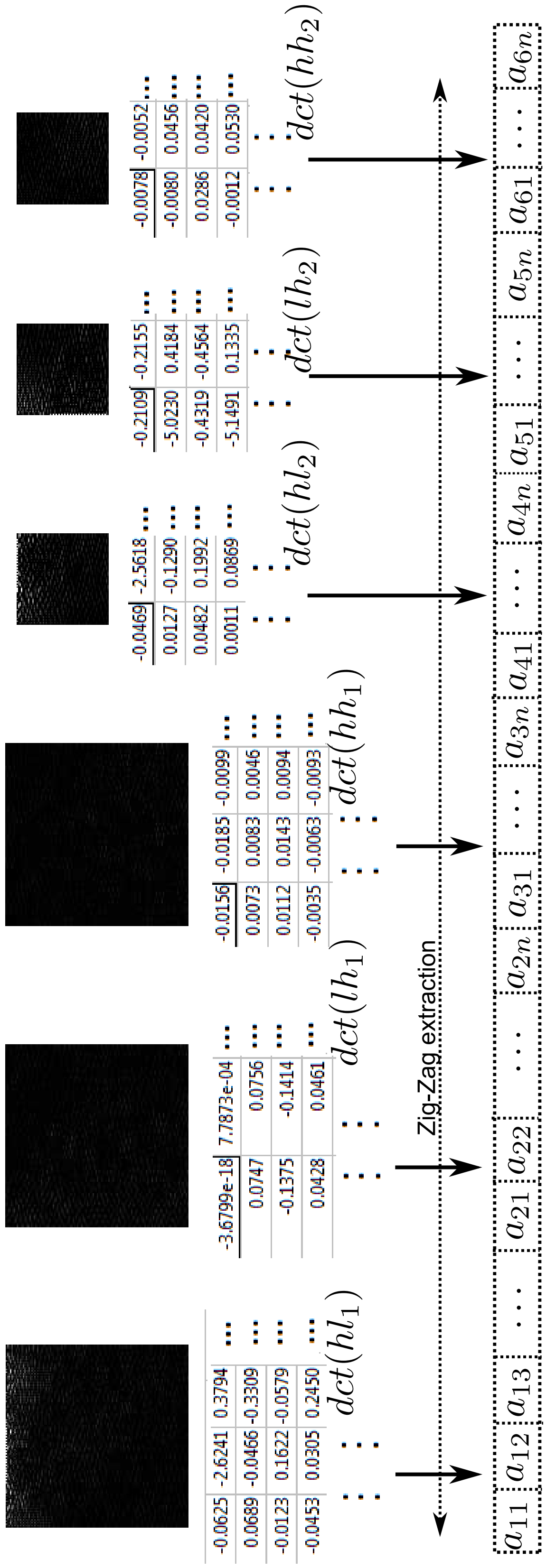


Fig. 5.3 – Selection of coefficients and construction of final feature vector.

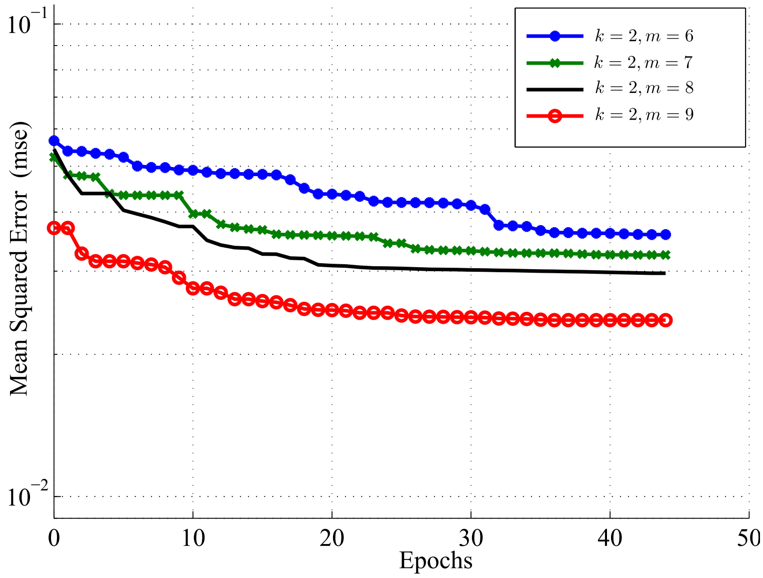


Fig. 5.4 – Comparison of convergence characteristics for level-2 decomposition with varied number of coefficients.

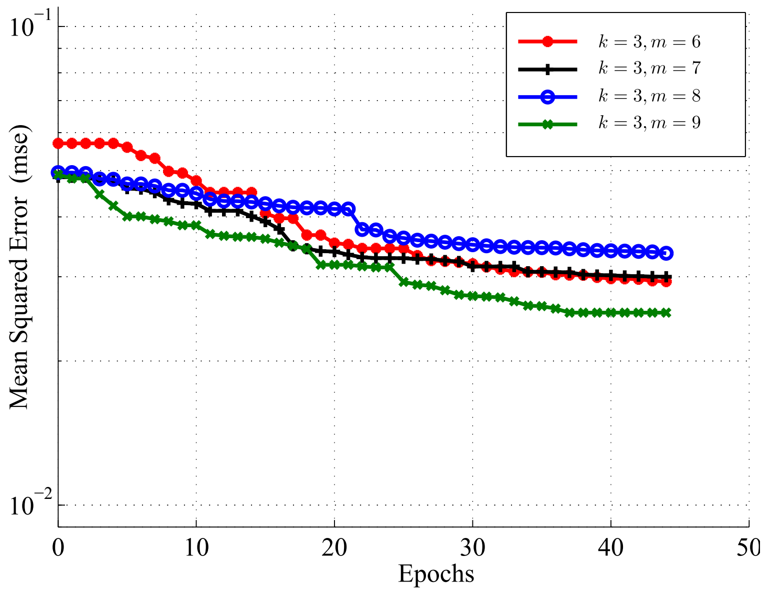


Fig. 5.5 – Comparison of convergence characteristics for level-3 decomposition with varied number of coefficients.

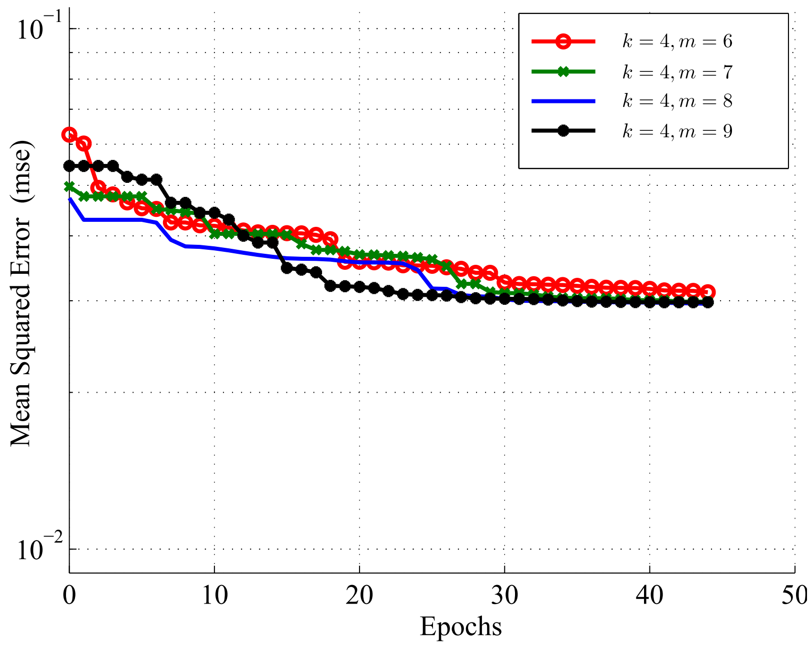


Fig. 5.6 – Comparison of convergence characteristics for level-4 decomposition with varied number of coefficients.

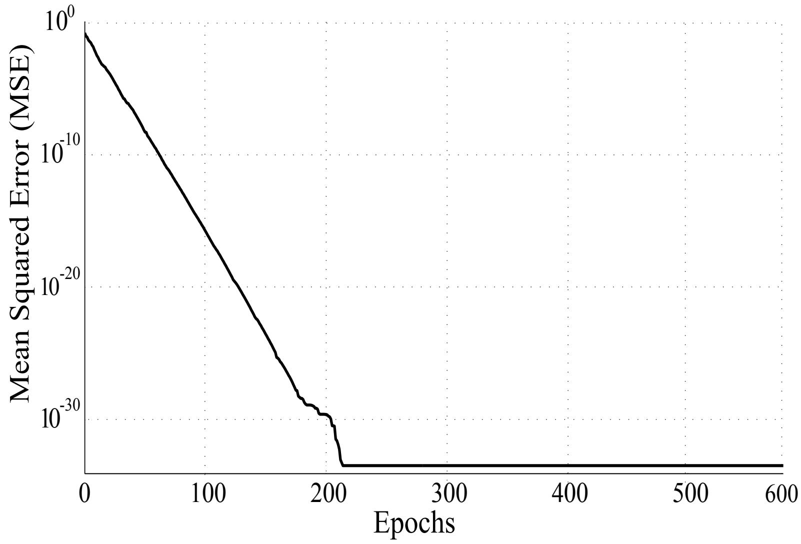


Fig. 5.7 – Convergence characteristics of BPNN with DCT features as input.

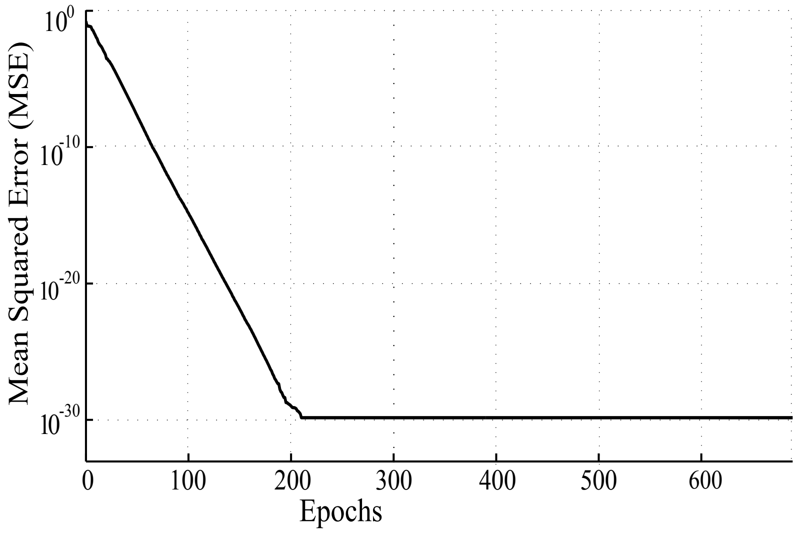


Fig. 5.8 – Convergence characteristics of BPNN with the DWT features as input.

5.2.1 Determining the Level of Decomposition and Number of Coefficients

For a pilot experiment with 150 training samples per class, we first choose the OHCS 1.0 dataset. Each individual character image is represented as a 256×256 gray scale image. The proposed feature is based on two transforms DWT and DCT, and both of the works on variable parameters set like, number of decomposition level for the DWT as well as number of DCT coefficients chosen from each sub-band decomposition. For example, for k level DWT decomposition we get $3*k$ high frequency sub-band images. DCT of these images are computed and m number of coefficients are chosen to form feature vector of length $3*k*m$. The value of m is varied through 6 to 9 in a step of 1. For each decomposition with varied coefficients, we train an individual BPNN classifier with $3*k*m$ input size. For suitable selection of k and m, the number of output neurons is fixed to 57 to classify fifty-seven different characters. Thus, for two levels of the DWT decomposition, we have four neural networks with 36, 42, 48, and 54 inputs respectively for 6, 7, 8, and 9 number of coefficients in a sub-band image. Similarly, different such neural networks are trained at different decomposition levels. The training convergence characteristics are shown in Figures 5.4, 5.5, and 5.6 for 2, 3, and 4 level of the DWT decompositions respectively with a varied number of coefficients. In each case, a same number of epochs are used for uniform comparison. It is observed that in each case, on increasing the number of coefficients the learning rate is increased, thus giving better recognition results. However, as the level of decomposition increased, the rate of increase in the learning rate gradually slows down. This implies that on increasing the number of coefficients from 8 to 9 in a $level - 2$ decomposition a drastic increase of about -5dB is observed while for the same case in $level - 3$ decomposition is observed to be about -1.3dB.

The best performance from each of the three cases individually is selected for further study. For selecting the suitable level of decomposition, a comparison is performed between the convergence plots with 8 coefficients in each level. As it can be observed from Fig. 5.9 that there is no significant change between the characteristics of level-3 and level-4. Thus choosing level-3 seems to be ideal while considering the computational overhead. From this pilot experiment, with $k = 3$ and $m = 8$, we get the best convergence characteristic. Hence, we select three levels of the DWT decomposition with 8 DCT coefficients for each character. The corresponding neural network is chosen as the classifier for the best dataset.

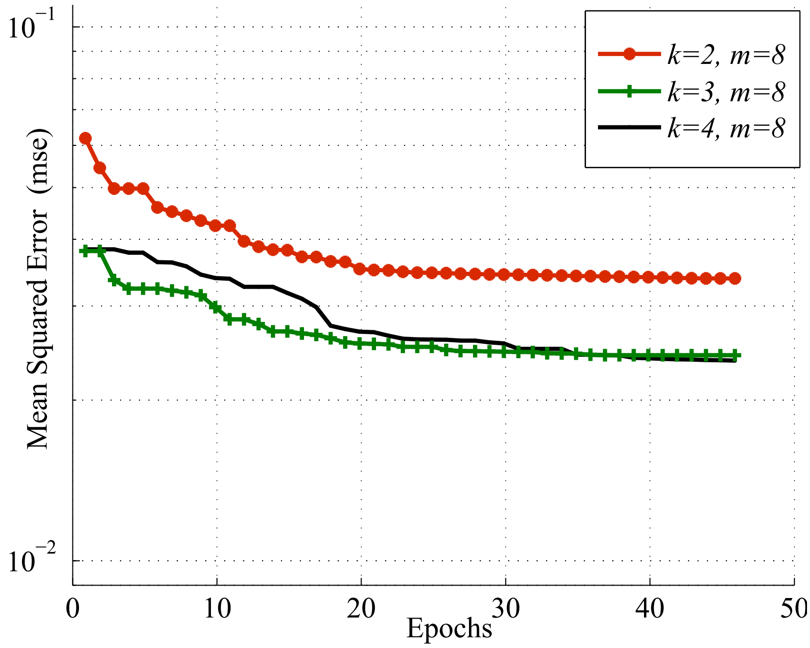
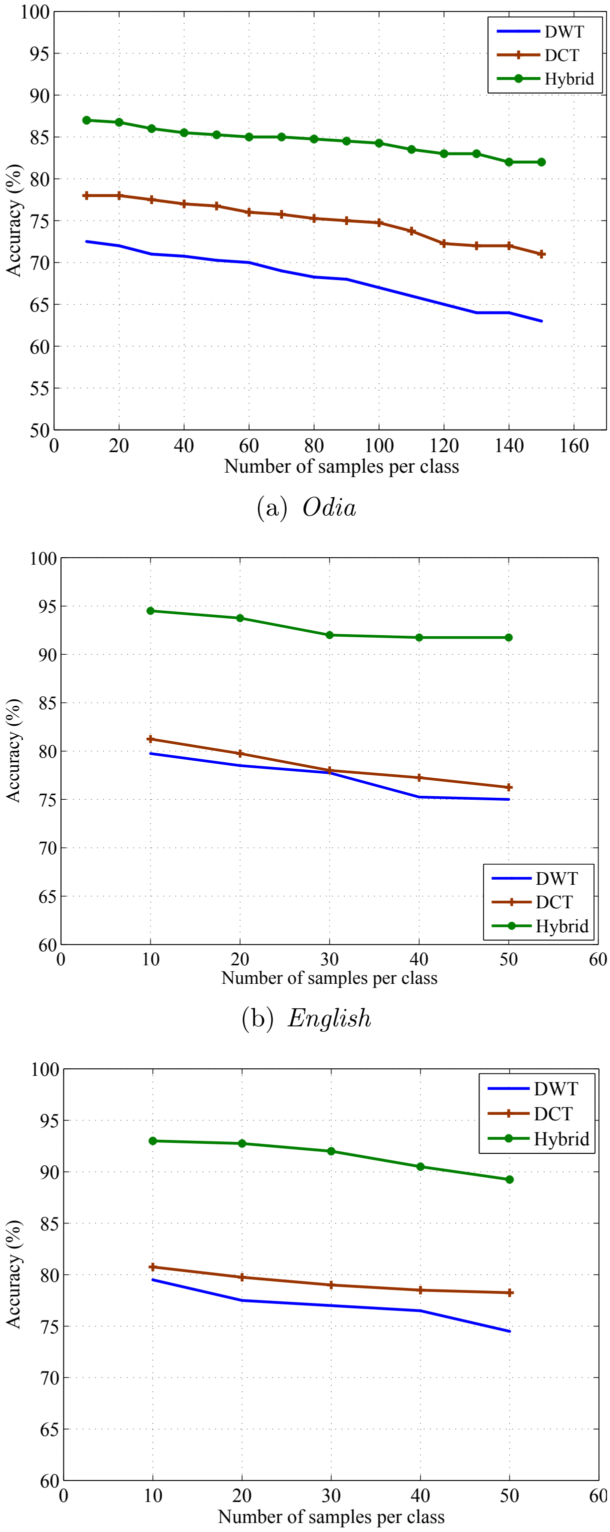


Fig. 5.9 – Convergence characteristics for different levels of decompositions.

5.3 Simulation and Results

After selecting the best parameters for the proposed HEFNC algorithm on a training set of 8,550 character images, the algorithm is tested against two state-of-the-art schemes namely DCT and DWT. A total of 72 ($m = 8$) DCT coefficients are extracted from each input test image from their DCT transforms, and they are subjected to classification using BPNN. For testing, a total of $8,550$ different samples (150 distinct samples from each of the 57 classes) of characters are considered. Three-level ($k = 3$) DWT is applied to the test images and all the corresponding feature values are used for classification using BPNN. The respective training convergence plots obtained using the DCT and DWT features Fig. 5.7, and Fig. 5.8. Similarly, to validate the HEFNC scheme, same 150 samples are taken into consideration. Hybrid energy features are extracted with $k = 3, m = 8$ and they are subjected to the corresponding classification using BPNN. The k-fold cross-validation ($k = 10$) technique is adopted for the purpose of computing overall accuracies for both the schemes separately. The samples are randomized and the entire dataset is used to determine the accuracy of the classification. Finally, the rates of recognition for the three languages are plotted distinctly for the DWT, DCT, and hybrid feature. These plots are shown in Fig. 5.10. Results show that the proposed algorithm works much better than the other two. The primary reason for this is lying in the fact that both filtering and compression techniques are applied to the images results in improved features with better image information. Implementing the DWT and DCT separately on the *Odia* dataset, we obtain overall training accuracy rates of 74.5% and 78.5% respectively, where as the proposed hybrid feature outperforms them with 86%.

Results suggest that the best recognition rate is achieved for 72 coefficients at $level - 3$ decomposition. Increasing the parameters does not lead to any significant increase in accuracy levels or learning rate.



(a) *Odia*

(b) *English*

(c) *Bangla*

Fig. 5.10 – Comparing the rates of training accuracy for the DWT, DCT, and HEFNC schemes for the three languages.

The proposed scheme is further implemented on the *English* and *Bangla* datasets.

Table 5.1 – Classification accuracy for *Odia*, *English*, and *Bangla* samples.

Sample Type	**Feature**	**Rate of Classification (%)**	
		Train	Test
Odia	SDBC scheme[84]	74	68.75
	TPID + ANN [87]	74.5	71.25
	GAT + ANN[90]	73.5	67.75
	DWT + ANN	74.5	68
	DCT + ANN	78.5	75
	CFNC scheme	88	80.25
	AFHMMC scheme	88.75	84.5
	ECLF-FC scheme	*92*	*88*
	HEFNC scheme	**87.75**	**86**
English	SDBC scheme	79	78.25
	TPID scheme	77	73
	GAT + ANN	75.5	71.5
	DWT + ANN	82	77.25
	DCT + ANN	83	78.5
	CFNC scheme	92	90.75
	AFHMMC scheme	93	92
	ECLF-FC scheme	*96.25*	*95*
	HEFNC scheme	**92.75**	**90.5**
Bangla	SDBC scheme	79.75	78
	TPID scheme	76.25	74
	GAT + ANN	75.5	71.5
	DWT + ANN	83	77
	DCT + ANN	84.75	79.25
	CFNC scheme	89	87.5
	AFHMMC scheme	95	89.5
	ECLF-FC scheme	*93.5*	*92*
	HEFNC scheme	**91.5**	**88.25**

The levels of decomposition and number of coefficients remain unchanged. From the *English* dataset, 50 samples from each of the 36 (alphabets + digits) classes are considered for training phase and remaining samples for the testing phase. Thus, the train/ test samples size so considered is 1,800/1,800. Similarly, from the *Bangla* dataset, we chose train/ test sample size to be 500/500. A comparison of overall rates of recognition of the proposed scheme with that of relevant schemes for all of these three languages is noted down in Table 5.1. It can be noticed that the hybrid feature performs well as compared to others, irrespective of the number of samples (language). Rates of misclassification using the ECLF-FC scheme is presented in Table 5.2 with a comparison to the proposed CFNC, AFHMMC, and ECLF-FC schemes. It can be observed that the HEFNC scheme is superior to others but inferior to ECLF-FC.

Table 5.2 – Rates of misclassification using different schemes for homogeneously shaped characters.

Class	Similar class	Misclassification rate (%)			
		CFNC scheme	AFVFHMMC scheme	ECLF-FC scheme	HEFNC scheme
ଅ ('ah')	ଥ ('tha')	22	21	16.5	17.5
ଥ ('tha')	ଅ ('ah')	14.5	12.5	11	12
ଇ ('ih')	ଉ ('uh')	10.5	10	6.5	8
ଇ ('ih')	ଲ ('la')	8	7.5	6	6
ଇ ('ih')	ର ('ra')	6.5	6.5	4	5
ଉ ('uh')	ଭ ('bha')	19	18	16	16.5
ଭ ('bha')	ର ('ra')	10	8	7.5	9
ଷ ('sha')	କ୍ଷ ('kshya')	12.5	10	6.5	11
୨ ('two')	୭ ('seven')	11.5	11	9	7
୨ ('two')	୬ ('six')	9	8.5	6.5	6
୬ ('six')	୭ ('seven')	24	22	15	17
୭ ('seven')	୬ ('six')	15	14	12	12.5
୭ ('seven')	୨ ('two')	12	11.5	10	11

5.4 Summary

In this chapter, we have proposed a hybrid feature based on the DWT and DCT. To decide about the optimal number of the DWT decomposition and number of DCT coefficients in each sub-band, exhaustive simulation has been performed. Based on the experimental results, a neural classifier is trained to be used during testing. An exhaustive test dataset is used and accuracy measures are compared along with existing schemes. It is in general observed that the proposed feature outperforms others. The scheme provides high accuracy and recognition rates and can be used as a general feature extractor for all purposes of image classification and recognition. As a part of the future work, we propose to extend this algorithm and implement it for compound characters of the *Odia* language.

Chapter 6

Conclusions and Future Work

The thesis deals with the proposition of novel features for handwritten *Odia* character recognition. Since for validation of any scheme, a public database for handwritten atomic *Odia* character set is not available, the author has tried to create a database of $17,100$ ($57 * 300$) samples collected from distinct users at two different times. It includes both of the 47 alphabets and 10 *Odia* numerals. The samples are collected using a digital note-maker.

The character images are standardized to a 256×256 size and subjected to conventional image pre-processing techniques like binarization, thinning, dilation, and enhancement to improve the quality. The database is made available online on *http://nitrkl.ac.in/Academic/Academic_ Centers/Data_ Computer_ Vision.aspx* for the use of researchers.

We have proposed four different features for recognition of handwritten *Odia* characters, namely

a. Contour-based primary features,

b. Aggregated contour features,

c. Evidence collection based features,

d. Hybrid energy features using DWT and DCT.

The first scheme divides contour of each character into thirty segments. Taking the centroid of the character as base point, three primary features length, angle, and chord-to-arc-ratio are extracted from each segment. Thus, there are 30 feature values for each primary attribute. For recognition, a back propagation neural network has been employed.

The second contribution falls in the line of feature reduction of the primary features. A fuzzy inference system has been employed to generate an aggregated feature vector of size 30 from 90 different feature points for each character. The key idea is to select one relevant feature point for each segment of the contour. For recognition, a 6-state HMM is employed for each character and as a consequence we have fifty-seven ergodic HMM's with 6-states each. The test character is passed through the HMM's and the log-likelihood measure is computed in each case. The maximum log-likelihood value identifies the label among the classes of characters. An accuracy of 84.5% has been achieved on our dataset.

The third contribution involves selection of evidence which are the most informative local shape contour features. A dedicated distance metric namely, far_count is used in computation of the information gain values for possible segments of different lengths that are extracted from whole shape contour of a character. The segment, with highest information gain value is treated as the evidence for it's corresponding class. An evidence dictionary is developed out of these evidence from all classes of characters and is used for testing purpose. An overall testing accuracy rate of 88% is obtained.

The final contribution deals with the development of a hybrid feature derived from DWT and DCT. Experimentally we have seen that a 3-level DWT decomposition with 72 DCT coefficients from each high-frequency components as features gives a testing accuracy of 86% in neural classifier.

The suggested features are studied in isolation and extensive simulations has been carried out. The competent schemes are also implemented on the same dataset. Further, studying generalization behaviour of proposed features, they are applied on *English* and *Bangla* datasets. The performance parameters like recognition rate misclassification rate are computed and compared. Further, as we progress from one contribution to the other, they are compared in each stage. In general, it is observed that the proposed features outperform the competent schemes and among all four suggested features local feature extraction based on evidence collection is better among other.

The thesis only deals with the recognition of atomic handwritten characters, It will be really challenging to extend them to compound characters. Our next thrust is active in this direction. Also, the features can be tested on different classifiers like SVM (support vector machine), SVR (support vector regression) etc.

Bibliography

[1] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21 – 27, 1967.

[2] J. Goin, "Classification bias of the k-nearest neighbor algorithm," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 379 – 381, 1984.

[3] P. Hart, "The condensed nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515 – 516, 1968.

[4] D. G. S. Richard O. Duda, Peter E. Hart, *Pattern Classification (Second Edition)*. Wiley, 2000.

[5] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkofp, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181 – 201, 2001.

[6] S. S. Haykin, *Neural Networks and Learning Machines*. Prentice Hall, 2009.

[7] J. Mantas, "An overview of character recognition methodologies," *Pattern Recognition*, vol. 19, no. 6, pp. 425 – 430, 1986.

[8] S. Mori, K. Yamamoto, and M. Yasuda, "Research on machine recognition of handprinted characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 4, pp. 386 – 405, 1984.

[9] T. S. El-Sheikh and R. M. Guindi, "Computer recognition of Arabic cursive scripts," *Pattern Recognition*, vol. 21, no. 4, pp. 293 – 302, 1988.

[10] C. Suen, M. Berthod, and S. Mori, "Automatic recognition of handprinted characters," vol. 68, no. 4, pp. 469 – 487, 1980.

[11] C. Tappert, C. Suen, and T. Wakahara, "The state of the art in online handwriting recognition," *Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, pp. 787 – 808, 1990.

[12] R. Bozinovic and S. Srihari, "Off-line cursive script word recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 68 – 83, 1989.

[13] V. Govindan and A. Shivaprasad, "Character recognition - a review," *Pattern Recognition*, vol. 23, no. 7, pp. 260 – 268, 1990.

[14] A. Belaid and J. P. Haton, "A syntactic approach for handwritten mathematical formula recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 105 – 111, 1984.

[15] Y. Ding, F. Kimura, Y. Miyake, and M. Shridhar, "Evaluation and improvement of slant estimation for handwritten words," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, ser. ICDAR, 1999, pp. 753 – 756.

[16] K. F. Chan and D. Y. Yeung, "Recognizing on-line handwritten alphanumeric characters through flexible structural matching," *Pattern Recognition*, vol. 32, no. 7, pp. 1099 – 1114, 1999.

[17] I. Ahmad, L. Rothacker, G. A. Fink, and S. A. Mahmoud, "Novel sub-character hmm models for Arabic text recognition," in *Proceedings of the International Conference on Document Analysis and Recognition, (ICDAR)*, 2013, pp. 658 – 662.

[18] I. Ahmad, G. Fink, and S. Mahmoud, "Improvements in sub-character hmm model based Arabic text recognition," in *Proceedings of Forteenth International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014, pp. 537 – 542.

[19] T. M. Breuel, "Segmentation of handprinted letter strings using a dynamic programming algorithm," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, 2001, pp. 821 – 826.

[20] S. W. Lee, D. J. Lee, and H. S. Park, "A new methodology for gray-scale character segmentation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1045 – 1050, 1996.

[21] S. Wachenfeld, H. U. Klein, and X. Jiang, "Recognition of screen-rendered text," in *Proceedings of the Eighteenth International Conference on Pattern Recognition*, vol. 2, 2006, pp. 1086 – 1089.

[22] O. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition-a survey," *Pattern Recognition*, vol. 29, no. 4, pp. 641 – 662, 1999.

[23] R. G. Casey and E. Lecolinet, "Feature extraction methods for character recognition-a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 690 – 706, 1996.

[24] T. Saba, A. Rehman, and M. Elarbi-Boudihir, "Methods and strategies on off-line cursive touched characters segmentation: a directional review," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 1047 – 1066, 2011.

[25] A. Dutta and S. Chaudhury, "Bengali alpha-numeric character recognition using curvature features," *Pattern Recognition*, vol. 26, no. 12, pp. 1757 – 1770, 1993.

[26] B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system," *Pattern Recognition*, vol. 31, no. 5, pp. 531 – 549, 1998.

[27] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, “A benchmark image database of isolated Bangla handwritten compound characters,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 4, pp. 413 – 431, 2014.

[28] U. Bhattacharya, M. Shridhar, S. Parui, P. Sen, and B. B. Chaudhuri, “Offline recognition of handwritten Bangla characters: an efficient two-stage approach,” *Pattern Analysis and Applications*, vol. 15, no. 4, pp. 445 – 458, 2012.

[29] S. Bag, G. Harit, and P. Bhowmick, “Recognition of Bangla compound characters using structural decomposition,” *Pattern Recognition*, vol. 47, no. 3, pp. 1187 – 1201, 2014.

[30] S. Bag, “A survey on optical character recognition for Bangla and Devanagari scripts,” *Sadhana Academy Proceedings in Engineering Sciences*, vol. 38, no. 1, pp. 133 – 168, 2013.

[31] I. K. Sethi and B. Chatterjee, “Machine recognition of hand-printed Devanagari numerals,” *Journal of the IETE*, vol. 22, no. 8, pp. 532 – 535, 1976.

[32] I. K. Sethi, “Machine recognition of constrained hand- printed Devanagari,” *Pattern Recognition*, vol. 9, no. 2, pp. 69 – 75, 1977.

[33] R. M. K. Sinha and H. N. Mahabala, “Machine recognition of Devanagari script,” *IEEE Transaction on systems, Man and Cybernetics*, vol. 9, no. 8, pp. 435 – 441, 1979.

[34] R. M. K. Sinha, “Role of context in Devanagari script recognition,” *Journal of the IETE*, vol. 33, no. 3, pp. 86 – 91, 1987.

[35] R. Karnik, “Identifying Devanagari characters,” in *Proceedings of Fifth International Conference on on Document Analysis and Recognition (ICDAR)*, 1999, pp. 669 – 672.

[36] V. Bansal and R. M. K. Sinha, “On how to describe shapes of Devanagari characters and use them for recognition,” in *Proceedings of Fifth International Conference on on Document Analysis and Recognition (ICDAR)*, 1999, pp. 410 – 413.

[37] K. Kale, P. D. S. Chavan, M. Kazi, and Y. Rode, “Zernike moment feature extraction for handwritten Devanagari (Marathi) compound character recognition,” *International Journal of Advanced Research in Artificial Intelligence*, vol. 3, no. 1, pp. 68 – 76, 2014.

[38] R. S. R. Kunte and R. S. Samuel, “On-line character recognition for handwritten Kannada characters using wavelet features and neural classifier,” *IETE Journal of Research*, vol. 46, no. 5, pp. 387 – 392, 2000.

[39] R. S. Kunte and R. S. Samuel, “A simple and efficient optical character recognition system for basic symbols in printed Kannada text,” *Sadhana, Academy Proceedings in Engineering Sciences*, vol. 32, no. 5, pp. 521 – 533, 2007.

[40] S. Niranjan, V. Kumar, G. Kumar, and V. Aradhya, “Fld based unconstrained handwritten Kannada character recognition,” in *Second International Conference on Future Generation Communication and Networking Symposia*, 2008, pp. 7 – 10.

[41] D. Shweta and S. Ramya, "Comparison of smoothing techniques and recognition methods for online Kannada character recognition system," in *International Conference on Advances in Engineering & Technology Research (ICAETR)*, 2014, pp. 1 – 5.

[42] G. S. G. Rajput and R. Horakeri, "Zone based handwritten Kannada character recognition using crack code and svm," in *International Conference on Advances in Computing, Communications and Informatics*, 2013, pp. 1817 – 1821.

[43] G. Siromoney and R. Chandrasekaran, "Computer recognition of printed Tamil characters," *Pattern Recognition*, vol. 10, no. 4, pp. 243 – 247, 1978.

[44] P. Chinnuswamy and S. Krishnamoorthy, "Recognition of handprinted Tamil characters," *Pattern Recognition*, vol. 12, no. 3, pp. 141 – 152, 1980.

[45] R. Kannan and R. Prabhakar, "An improved handwritten Tamil character recognition system using octal graph," *Journal of Computer Sciences*, vol. 4, no. 7, pp. 509 – 516, 2008.

[46] S. Raja and M. John, "A novel Tamil character recognition using decision tree classifier," *IETE Journal of Research*, vol. 59, no. 5, pp. 569 – 575, 2013.

[47] A. Subashini and N. Kodikara, "A novel sift-based codebook generation for handwritten Tamil character recognition," in *IEEE Sixth International Conference on Industrial and Information Systems (ICIIS 2011)*, 2011, pp. 261 – 264.

[48] S. Mohanty, H. Dasbebartta, and T. Behera, "An efficient bilingual optical character recognition (English-Oriya) system for printed documents," in *Proceedings of the Seventh International Conference on Advances in Pattern Recognition (ICAPR)*, 2009, pp. 398 – 401.

[49] S. Nigam and A. Khare, "Multifont Oriya character recognition using curvelet transform," in *Communications in Computer and Information Science*, vol. 139, 2009, pp. 544 – 550.

[50] A. Rao, "Multifont Oriya character recognition using curvelet transform," in *Proceedings of the International Conference on Imaging Science, Systems, and Technology (CISST)*, vol. 1, 2000, pp. 245 – 251.

[51] T. Wakabayashi, U. Pal, and F. Kimura, "F-ratio based weighted feature extraction for similar shape character recognition," in *Proceedings of the Tenth International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 196 – 200.

[52] K. Roy, U. Pal, and F. Kimura, "Oriya handwritten numeral recognition system," in *Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR)*, 2005, pp. 770 – 774.

[53] N. Tripathy and U. Pal, "System for Oriya handwritten numeral recognition," in *Proceedings SPIE - The International Society for Optical Engineering*, vol. 5296, 2004, pp. 174–181.

[54] http://www.isical.ac.in/ ujjwal/download/oriyanumeral.html.

[55] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (Third Edition).* Prentice-Hall, Inc., 2006.

[56] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the Forteenth International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'95. Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143.

[57] http://www.iam.unibe.ch/fki/databases.

[58] http://www.isical.ac.in/ ujjwal/download/numeral-online.html.

[59] U. Pal, T. Wakabayashi, and F. Kimura, "A system for off-line Oriya handwritten character recognition using curvature feature," in *Proceedings of the Tenth International Conference on Information Technology, ICIT*, 2007, pp. 227 – 229.

[60] T. Wakabayashi, U. Pal, F. Kimura, and Y. Miyake, "F-ratio based weighted feature extraction for similar shape character recognition," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2009, pp. 196–200.

[61] S. Bag, P. Bhowmick, and G. Harit, "Recognition of Bengali handwritten characters using skeletal convexity and dynamic programming," *International Conference on Emerging Applications of Information Technology*, pp. 265 – 268, 2011.

[62] V. Ghods, E. Kabir, and F. Razzazi, "Decision fusion of horizontal and vertical trajectories for recognition of online Farsi subwords," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 544 – 550, 2013.

[63] E. B. S., "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 767 – 779, 2011.

[64] J. Wang and R. Zhu, "Handwritten Nushu character recognition based on hidden markov model," *Journal of Computers*, vol. 5, no. 5, pp. 663 – 670, 2010.

[65] H. A. Al-Muhtaseb, S. A. Mahmoud, and R. S. Qahwaji, "Recognition of off-line printed Arabic text using hidden markov models," *Signal Processing*, vol. 88, no. 12, pp. 2902 – 2912, 2008.

[66] S. Wang, S. Uchida, M. Liwicki, and Y. Feng, "Part-based methods for handwritten digit recognition," *Frontiers of Computer Science*, vol. 7, no. 4, pp. 514 – 525, 2013.

[67] T. Artieres, S. Marukatat, and P. Gallinari, "Online handwritten shape recognition using segmental hidden markov models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 205 – 217, 2007.

[68] T. K. Bhowmik, S. K. Parui, U. Bhattacharya, and B. Shaw, "An hmm based recognition scheme for handwritten Oriya numerals," in *Proceedings of the Ninth International Conference on Information Technology*, ser. ICIT '06. IEEE Computer Society, 2006, pp. 105 –110.

[69] P. R. Cavalin, R. Sabourin, C. Y. Suen, and A. S. B. Jr., "Evaluation of incremental learning algorithms for hmm in the recognition of alphanumeric characters," *Pattern Recognition*, vol. 42, no. 12, pp. 3241 – 3253, 2009.

[70] M. I. Razzak, F. Anwar, S. Husain, A. Belaid, and M. Sher, "Hmm and fuzzy logic: A hybrid approach for online Urdu script-based languages character recognition," *Knowledge-Based Systems*, vol. 23, no. 8, pp. 914 – 923, 2010.

[71] W. Zeng, X. Meng, C. Yang, and L. Huang, "Feature extraction for online handwritten characters using delaunay triangulation," *Computers & Graphics*, vol. 30, no. 5, pp. 779 – 786, 2006.

[72] T. Theeramunkong and C. Wongtapan, "Off-line isolated handwritten Thai OCR using island-based projection with n-gram model and hidden markov models," *Information Processing & Management*, vol. 41, no. 1, pp. 139 – 160, 2005.

[73] R. B. Milan Sonka, Vaclav Hlavac, *Image Processing: Analysis and Machine Vision*, 2nd ed. Cengage Learning, 2008.

[74] L. R. Rabiner, "Readings in speech recognition." Morgan Kaufmann Publishers Inc., 1990, ch. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 267 – 296.

[75] G. Fink and T. PlÃűtz, "Developing pattern recognition systems based on markov models: The esmeralda framework," *Pattern Recognition and Image Analysis*, vol. 18, no. 2, pp. 207 – 215, 2008.

[76] O. Perez, M. Piccardi, J. Garcia, M. Patricio, and J. Molina, "Comparison between genetic algorithms and the baum-welch algorithm in learning hmms for human activity classification," in *Applications of Evolutionary Computing*, ser. Lecture Notes in Computer Science, vol. 4448. Springer Berlin Heidelberg, 2007, pp. 399 – 406.

[77] D. M. S. Abirami, "Local features-based script recognition from printed bilingual document images," *International Journal of Computer Applications in Technology (IJCAT)*, vol. 38, no. 4, pp. 283 – 297, 2010.

[78] R. Brown, T. Fay, and C. Walker, "Handprinted symbol recognition system," *Pattern Recognition*, vol. 21, no. 2, pp. 91 – 118, 1988.

[79] L. Shaozi, C. Donglin, and C. Shuyuan, "Document image classification based on structured local edge pattern," *Journal of Xiamen University (Natural Science)*, vol. 52, no. 3, pp. 349 – 355, 2013.

[80] D. Tao, L. Liang, L. Jin, and Y. Gao, "Similar handwritten Chinese character recognition by kernel discriminative locality alignment," *Pattern Recognition Letters*, vol. 35, no. 1, pp. 186 – 194, 2014.

[81] A. Zaafouri, M. Sayadi, and F. Fnaiech, "Printed Arabic character recognition using local energy and structural features," in *Proceedings of the Second International Conference on Communications Computing and Control Applications, CCCA 2012*, 2012, pp. 1 – 5.

[82] K. S. Daryoush, M. Khademi, A. Nikookar, and A. Farahani, “The application of local linear neuro fuzzy model in recognition of online persian isolated characters,” in *Proceedings of the Third International Conference on Advanced Computer Theory and Engineering, ICACTE 2010*, 2010, pp. 5574 – 5577.

[83] Y. Jin, K. Qiu, Y. Dai, G. Xiao, and H. Deng, “An improved handwritten Chinese character recognition based on localized ellipse model,” in *Proceedings of the Third International Congress on Image and Signal Processing, CISP 2010*, 2010, pp. 1803 – 1807.

[84] F. Yu, L. Wen-ju, S. Juan-hong, Z. Ying, and S. Jia-wei, “Chinese character recognition of license plate based on wavelet transform and fractal dimension,” *Computer Engineering*, vol. 37, no. 22, pp. 137 – 138, 2011.

[85] C. Li and D. Xiao-qing, “Font recognition of single Chinese character based on wavelet feature,” *Acta Electronica Sinica*, vol. 32, no. 2, pp. 177 – 180, 2004.

[86] Y. Ding, F. Yiping, and L. Zhineng, “Fast character recognition algorithm based on wavelet transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 317 – 321, 2004.

[87] M. Suzuki, N. Kato, H. Aso, and Y. Nemoto, “A handprinted character recognition system using image transformation based on partial inclination detection,” *IEICE Transactions on Information and Systems*, vol. E79-D, no. 5, pp. 504 – 509, 1996.

[88] S. Prasad, B. Singh, and P. Kumar, “Basic handwritten character recognition from multi-lingual image dataset using multi-resolution and multi-directional transform,” *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 10, no. 5, p. 1250046 (28 pp.), 2012.

[89] Y. Ying, Y. Yang, D. Cai-lin, H. Xiu-ling, and C. Zeng-zhao, “Application of wavelet packet transformation in handwritten amount Chinese characters recognition,” *Computer Engineering and Applications*, vol. 44, no. 31, pp. 229 – 232, 2008.

[90] T. Wakahara, Y. Kimura, and A. Tomono, “Affine-invariant recognition of gray-scale characters using global affine transformation correlation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 384 – 395, 2001.

[91] N. Ahmed, T. Natarajan, and K. Rao, “Discrete cosine transform,” *Computers, IEEE Transactions on*, vol. C-23, no. 1, pp. 90 – 93, 1974.

[92] F. Bayer and R. Cintra, “Image compression via a fast dct approximation,” *Latin America Transactions, IEEE (Revista IEEE America Latina)*, vol. 8, no. 6, pp. 708 – 713, 2010.

[93] C. Cheng and K. Parhi, “Hardware efficient fast dct based on novel cyclic convolution structures,” *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4419 – 4434, 2006.

[94] H. Huang and L. Xiao, “Cordic based fast radix-2 dct algorithm,” *Signal Processing Letters, IEEE*, vol. 20, no. 5, pp. 483 – 486, 2013.

[95] A. Karoui and R. Vaillancourt, "Families of biorthogonal wavelets," *Computers & Mathematics with Applications*, vol. 28, no. 4, pp. 25 – 39, 1994.

[96] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 247 – 269, 1998.

[97] A. Graps, "An introduction to wavelets," *Computational Science Engineering, IEEE*, vol. 2, no. 2, pp. 50 – 61, 1995.

[98] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511 – 546, 1998.

[99] J. Kovacevic and W. Sweldens, "Wavelet families of increasing order in arbitrary dimensions," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 480 – 496, 2000.

Dissemination

Published

1. **Tusar Kanti Mishra**, Banshidhar Majhi, and Pankaj K Sa. Model based Odia numeral recognition using fuzzy aggregated features. *Frontiers of Computer Science (Springer)*, Volume 8, Issue 6, pages 916 – 922, 2013.

2. **Tusar Kanti Mishra**, Banshidhar Majhi, and Ratnakar Dash. Shape Descriptors-based Generalized Scheme for Handwritten Character Recognition. *International Journal of Computer Vision and Robotics (Inderscience)*, In press, 2015.

3. **Tusar Kanti Mishra**, Banshidhar Majhi, and Ratnakar Dash. A Contour Descriptors-Based Generalized Scheme for Handwritten Odia Numerals Recognition. *Journal of Information Processing Systems (JIPS)*, In press.

4. **Tusar Kanti Mishra**, Banshidhar Majhi, and Sandeep Panda. A comparative analysis of image transformations for handwritten Odia numeral recognition. In *IEEE International Conference on Informatics (ICACCI)*, Pages 790-793, Mysore, India, 2013.

5. Olarik Surinta, **Tusar Kanti Mishra**, Mahir F. Karaaba, Marco Wiering and Lambert R.B. Schomaker. Recognizing Handwritten Characters with Local Descriptors and Bags of Visual Words. In *Engineering Applications of Neural Networks (EANN)*, , Island of Rhodes, Greece, 2015.

BIO-DATA

Tusar Kanti Mishra

Date of Birth: 07 October 1982

Correspondence:
Department of Computer Science and Engineering,
National Institute of Technology Rourkela,
Rourkela – 769 008, Odisha, India.

Ph: +91 94386 25213

e-mail: tusar.k.mishra@gmail.com

Qualification

- M.Tech (CS)
  SUIIT, Sambalpur University [First class]
- B.Tech (CSE)
  BPUT, Odisha [First class with honors]

Professional Experience

Teaching Assistant, NIT Rourkela, Jan 2012 – Date

Erasmus mundus research scholar, RUG, The Netherlands, Sept 2014 – Feb 2015

Lecturer, Silicon Institute of Technology, July 2011 – Dec 2011

Lecturer, PKA College of Engineering, Sept 2004 – July 2011

Publications

- 04 Journal Articles
- 03 Conference Articles
- 01 Book Chapter

Permanent Address

C/O: Sri Narasingh Prasad Mishra, Sri Ram Vihar, Govindtola, Dhanupali, Sambalpur, Odisha - 768005.